Lexicographical Generation of a Generalized Dyck Language

Jens Liebehenschel
Johann Wolfgang Goethe-Universität, Frankfurt am Main
Fachbereich Informatik
D-60054 Frankfurt am Main, Germany
e-mail: jens@sads.informatik.uni-frankfurt.de

Abstract

Given two disjoint alphabets $T_{[}$ and $T_{]}$ and a relation $\mathcal{R} \subseteq T_{[} \times T_{]}$, the generalized Dyck language $D^{\mathcal{R}}$ over $T_{[} \cup T_{]}$ consists of all words $w \in (T_{[} \cup T_{]})^{\star}$ which are equivalent to the empty word ε under the congruence δ defined by $x\,y \equiv \varepsilon \bmod \delta$ for all $(x, y) \in \mathcal{R}$. In this paper we present an algorithm that generates all words of length $2n$ of the generalized Dyck language lexicographically. Thereby, each Dyck word is computed from its predecessor according to the lexicographical order without any knowledge about the Dyck words generated before. Additionally, we introduce a condition on the relation $\mathcal{R}$ for the language to be simply generated, which means that an algorithm needs to read only the suffix to be changed in order to compute the successor of a word according to the lexicographical order. Furthermore, we analyze the algorithm that generates the Dyck words. For arbitrary $\mathcal{R}$, we compute the s-th moments of the random variable describing the length of the suffix to be changed in the computation of the successor of a Dyck word according to the lexicographical order.

Keywords: Dyck language, lexicographical generation, average-case analysis.

1 Overview and Definitions

In this section we introduce the generalization of the Dyck language, the lexicographical order needed for the lexicographical generation and present all definitions – illustrated by several examples – for the whole paper. Further, we point out the contents of the following sections.

In this paper we present an algorithm that generates all words of length $2n$ of the generalized Dyck language given in Definition 1 lexicographically. The algorithm reads a word from right to left and changes a suffix of that word in order to generate the next word according to the lexicographical order given in Definition 2.

Definition 1:
Let $t_1, t_2 \in \mathbb{N}$ and $T_{[} := \{[_1, [_2, \ldots, [_{t_1}\}$ (resp. $T_{]} := \{]_1,]_2, \ldots,]_{t_2}\}$) be the *set of opening* (resp. *closing*) *brackets*. Let $|S|$ be the cardinality of the set S, so $|T_{[}| = t_1$ and $|T_{]}| = t_2$. With $T := T_{[} \,\dot\cup\, T_{]}$, where $\dot\cup$ denotes the disjoint union of sets, and a relation $\mathcal{R} \subseteq T_{[} \times T_{]}$ we obtain the *generalized Dyck language associated with* $\mathcal{R}$ by

$$D^{\mathcal{R}} := \{w \in T^{\star} \mid w \equiv \varepsilon \bmod \delta\} ,$$

where ε denotes the *empty word* and δ is the congruence over T which is defined by $(\forall([_a,]_b) \in \mathcal{R})([_a\,]_b \equiv \varepsilon \bmod \delta)$. Let $\mathsf{length}(w)$ be the *number of symbols of the word* w. Obviously, $\mathsf{length}(\varepsilon) = 0$. The *set of all Dyck words of length* $2n$ is given by $D^{\mathcal{R}}_{2n} := \{w \in D^{\mathcal{R}} \mid \mathsf{length}(w) = 2n\}$. □

Remark 1:
Obviously, there is a unique *corresponding closing* (resp. *opening*) *bracket* to each opening (resp. closing) bracket in every Dyck word $w \in D^{\mathcal{R}}$.

The corresponding closing bracket to an opening bracket in a Dyck word $w \in D^{\mathcal{R}}$ can be found by searching the first closing bracket behind the shortest word $w_2 \in$

$D^{\mathcal{R}}$ (w_2 might be ε) on the right side of the opening bracket. If the opening (resp. its corresponding closing) bracket is $[_a$ (resp. $]_b$), we have $w = w_1\,[_a\,w_2\,]_b\,w_3$ with $w_2, w_1\,w_3 \in D^{\mathcal{R}}$. The corresponding opening bracket to a closing bracket can be found in an analogous way.

Definition 2:
Let $<_{\text{lex}} \subseteq T \times T$ be an irreflexive linear ordering on T. The *lexicographical order* $\prec_{\text{lex}}$ *over* T^+ is defined as the extension of $<_{\text{lex}}$ to $\prec_{\text{lex}} \subseteq T^+ \times T^+$ by $x \prec_{\text{lex}} y :\Leftrightarrow (\exists z \in T^+)(xz = y) \vee \big(\exists\,(w, x', y', a, b) \in T^{\star 3} \times T^2\big)(x = wax' \wedge y = wby' \wedge a <_{\text{lex}} b)$ [Ke98]. Note that in this paper we consider the lexicographical order on words of length $2n$ only. □

We use the following ordering on T:

$$[_{|T_{[}|} \;<_{\text{lex}}\; \cdots \;<_{\text{lex}}\; [_1 \;<_{\text{lex}}\;]_1 \;<_{\text{lex}}\; \cdots \;<_{\text{lex}}\;]_{|T_{]}|} \;.$$

Now, let us have a closer look at the relation $\mathcal{R}$. From the existence of the lexicographical order $\prec_{\text{lex}}$ on $D^{\mathcal{R}}$ results the existence of a unique *lexicographically minimal* (resp. *maximal*) *tuple in* $\mathcal{R}$ which is denoted by $\mathcal{R}_{\min}$ (resp. $\mathcal{R}_{\max}$).

Definition 3:
Let $p^{\mathcal{R}}_{\min}$ (resp. $p^{\mathcal{R}}_{\max}$) be the *lexicographically minimal* (resp. *maximal*) *pair of brackets in R* . Further, let $[_{p^{\mathcal{R}}_{\min}}, [_{p^{\mathcal{R}}_{\max}} \in T_{[}$ and $]_{p^{\mathcal{R}}_{\min}},]_{p^{\mathcal{R}}_{\max}} \in T_{]}$. Then we get:

$$\begin{aligned} p^{\mathcal{R}}_{\min} &:= [_{p^{\mathcal{R}}_{\min}}\,]_{p^{\mathcal{R}}_{\min}} \iff \mathcal{R}_{\min} = \big([_{p^{\mathcal{R}}_{\min}},]_{p^{\mathcal{R}}_{\min}}\big) \text{ and} \\ p^{\mathcal{R}}_{\max} &:= [_{p^{\mathcal{R}}_{\max}}\,]_{p^{\mathcal{R}}_{\max}} \iff \mathcal{R}_{\max} = \big([_{p^{\mathcal{R}}_{\max}},]_{p^{\mathcal{R}}_{\max}}\big)\;. \end{aligned}$$

Obviously, $[_{p^{\mathcal{R}}_{\min}}$ (resp. $]_{p^{\mathcal{R}}_{\min}}$) is the *opening* (resp. *closing*) *bracket of the lexicographically minimal pair of brackets* and $[_{p^{\mathcal{R}}_{\max}}$ (resp. $]_{p^{\mathcal{R}}_{\max}}$) is the *opening* (resp. *closing*) *bracket of the lexicographically maximal pair of brackets.* Now, we are able to compute the *minimal* (resp. *maximal*) *word of* $D^{\mathcal{R}}_{2n}$ *according to the lexicographical order* $w^{D^{\mathcal{R}}_{2n}}_{\min}$ (resp. $w^{D^{\mathcal{R}}_{2n}}_{\max}$):

$$w^{D^{\mathcal{R}}_{2n}}_{\min} = \big([_{p^{\mathcal{R}}_{\min}}\big)^n \big(]_{p^{\mathcal{R}}_{\min}}\big)^n \quad \text{and} \quad w^{D^{\mathcal{R}}_{2n}}_{\max} = \big(p^{\mathcal{R}}_{\max}\big)^n \;.$$

□

Definition 4:
Given $([_a,]_b) \in \mathcal{R}$, $]_b$ is called *minimal* (resp. *maximal*), if $(\forall c \in [1 : b-1])(([_a,]_c) \notin \mathcal{R})$ $\big($resp. $(\forall c \in [b+1 : |T_{]}|])(([_a,]_c) \notin \mathcal{R})\big)$. Here, $[a : b] := \{a, \ldots, b\}$ denotes the *set of all integers i with* $a \leq i \leq b$. □

Note that the property of being minimal (resp. maximal) depends not only on the closing bracket, but also on the corresponding opening bracket. For example, $]_b$ of $([_{a_1},]_b)$ can be minimal, even if $]_b$ of $([_{a_2},]_b)$ is not minimal.

In this paper we assume that $(\forall x \in T)(\exists([_a,]_b) \in \mathcal{R})(x = [_a \vee x =]_b)$, i. e. there are no useless symbols in T, so we find $[_{p^{\mathcal{R}}_{\min}} = [_{|T_{[}|}$ and $[_{p^{\mathcal{R}}_{\max}} = [_1$.

Let us demonstrate the above definitions by two simple examples.

Example 1:
Let $T := \{[_1, [_2,]_1,]_2\}$, $\mathcal{R} := \{([_1,]_1), ([_2,]_2)\}$ and $n := 2$. We get the lexicographical order $[_2 <_{\text{lex}} [_1 <_{\text{lex}}]_1 <_{\text{lex}}]_2$. The lexicographically minimal (resp. maximal) word is given by $w^{D^{\mathcal{R}}_4}_{\min} = [_2\,[_2\,]_2\,]_2$ $\big($resp. $w^{D^{\mathcal{R}}_4}_{\max} = [_1\,]_1\,[_1\,]_1\big)$. In Figure 1 we find all Dyck words of this example.

Example 2:
Let $T := \{[_1, [_2,]_1,]_2\}$, $\mathcal{R} := \{([_1,]_1), ([_1,]_2), ([_2,]_2)\}$ and $n := 2$. Again, we obtain the lexicographical order $[_2 <_{\text{lex}} [_1 <_{\text{lex}}]_1 <_{\text{lex}}]_2$. The lexicographically minimal (resp. maximal) word is given by $w^{D^{\mathcal{R}}_4}_{\min} = [_2\,[_2\,]_2\,]_2$ $\big($resp. $w^{D^{\mathcal{R}}_4}_{\max} = [_1\,]_2\,[_1\,]_2\big)$. Given

$$[_2\,[_2\,]_2\,]_2 \prec_{\mathrm{lex}} [_2\,[_1\,]_1\,]_2 \prec_{\mathrm{lex}} [_2\,]_2\,[_2\,]_2 \prec_{\mathrm{lex}} [_2\,]_2\,[_1\,]_1 \prec_{\mathrm{lex}}$$
$$[_1\,[_2\,]_2\,]_1 \prec_{\mathrm{lex}} [_1\,[_1\,]_1\,]_1 \prec_{\mathrm{lex}} [_1\,]_1\,[_2\,]_2 \prec_{\mathrm{lex}} [_1\,]_1\,[_1\,]_1$$

Figure 1: All Dyck words of Example 1 arranged according to the lexicographical order $\prec_{\mathrm{lex}}$.

$([_1,]_1)$, we see that $]_1$ is minimal, but not maximal. Analogously, $]_2$ of $([_1,]_2)$ is maximal, but not minimal. Regarding $([_2,]_2)$ we find $]_2$ to be both minimal and maximal. In Figure 2 all Dyck words of this example are arranged lexicographically.

$$[_2\,[_2\,]_2\,]_2 \prec_{\mathrm{lex}} [_2\,[_1\,]_1\,]_2 \prec_{\mathrm{lex}} [_2\,[_1\,]_2\,]_2 \prec_{\mathrm{lex}} [_2\,]_2\,[_2\,]_2 \prec_{\mathrm{lex}} [_2\,]_2\,[_1\,]_1 \prec_{\mathrm{lex}} [_2\,]_2\,[_1\,]_2 \prec_{\mathrm{lex}}$$
$$[_1\,[_2\,]_2\,]_1 \prec_{\mathrm{lex}} [_1\,[_2\,]_2\,]_2 \prec_{\mathrm{lex}} [_1\,[_1\,]_1\,]_1 \prec_{\mathrm{lex}} [_1\,[_1\,]_1\,]_2 \prec_{\mathrm{lex}} [_1\,[_1\,]_2\,]_1 \prec_{\mathrm{lex}} [_1\,[_1\,]_2\,]_2 \prec_{\mathrm{lex}}$$
$$[_1\,]_1\,[_2\,]_2 \prec_{\mathrm{lex}} [_1\,]_1\,[_1\,]_1 \prec_{\mathrm{lex}} [_1\,]_1\,[_1\,]_2 \prec_{\mathrm{lex}} [_1\,]_2\,[_2\,]_2 \prec_{\mathrm{lex}} [_1\,]_2\,[_1\,]_1 \prec_{\mathrm{lex}} [_1\,]_2\,[_1\,]_2$$

Figure 2: All Dyck words of Example 2 arranged according to the lexicographical order $\prec_{\mathrm{lex}}$.

Now, let us recall two definitions of [Ke98].
Definition 5:
For arbitrary language L and $w \in L$, $\mathsf{next}^L_{\prec_{\mathrm{lex}}}(w)$ is defined by

$$\mathsf{next}^L_{\prec_{\mathrm{lex}}}(w) := \begin{cases} w' & \text{if} \quad w \neq w^L_{\max} \wedge w \prec_{\mathrm{lex}} w' \wedge \\ & \qquad (\forall w'' \in L \setminus \{w, w'\})(w'' \prec_{\mathrm{lex}} w \vee w' \prec_{\mathrm{lex}} w'') \\ \text{undefined} & \text{if} \quad w = w^L_{\max} \end{cases}$$

and it is called *successor function* for the language L. □
Definition 6:
Let Σ be a finite alphabet, $L \subseteq \Sigma^\star$ be a formal language with the lexicographical order $\prec_{\mathrm{lex}}$. The functions $\mathsf{pre}^L_{\prec_{\mathrm{lex}}}, \mathsf{old}^L_{\prec_{\mathrm{lex}}}, \mathsf{new}^L_{\prec_{\mathrm{lex}}} : L \to \Sigma^\star$ are defined as follows:

$$\begin{aligned} \mathsf{pre}^L_{\prec_{\mathrm{lex}}}(w) &:= \begin{cases} \varepsilon & \text{if} \quad w = w^L_{\max} \\ u & \text{if} \quad \big(\exists (v, v') \in \Sigma^\star \times \Sigma^\star\big)\big(w = u\,v \wedge \\ & \qquad \mathsf{next}^L_{\prec_{\mathrm{lex}}}(w) = u\,v' \wedge \mathsf{length}(u) \text{ is maximal}\big) \end{cases}, \\ \mathsf{old}^L_{\prec_{\mathrm{lex}}}(w) &:= v \iff w = \mathsf{pre}^L_{\prec_{\mathrm{lex}}}(w)\,v\ , \\ \mathsf{new}^L_{\prec_{\mathrm{lex}}}(w) &:= \begin{cases} v & \text{if} \quad w \neq w^L_{\max} \wedge \mathsf{next}^L_{\prec_{\mathrm{lex}}}(w) = \mathsf{pre}^L_{\prec_{\mathrm{lex}}}(w)\,v \\ \text{undefined} & \text{if} \quad w = w^L_{\max} \end{cases}. \end{aligned}$$

The language L is called *simply generated* with respect to $\prec_{\mathrm{lex}}$ iff it satisfies the property

$$\big(\forall (w, w') \in L^2\big)$$
$$\big(\mathsf{old}^L_{\prec_{\mathrm{lex}}}(w) = \alpha\,\mathsf{old}^L_{\prec_{\mathrm{lex}}}(w'), \alpha \in \Sigma^\star \rightsquigarrow \big(\alpha, \mathsf{new}^L_{\prec_{\mathrm{lex}}}(w)\big) = \big(\varepsilon, \mathsf{new}^L_{\prec_{\mathrm{lex}}}(w')\big)\big)\ . \quad □$$

Obviously, $\mathsf{pre}^L_{\prec_{\mathrm{lex}}}(w)$ is the longest common prefix of w and $\mathsf{next}^L_{\prec_{\mathrm{lex}}}(w)$; $\mathsf{old}^L_{\prec_{\mathrm{lex}}}(w)$ (resp. $\mathsf{new}^L_{\prec_{\mathrm{lex}}}(w)$) is the suffix of w (resp. $\mathsf{next}^L_{\prec_{\mathrm{lex}}}(w)$) to the right of that longest common prefix of w and $\mathsf{next}^L_{\prec_{\mathrm{lex}}}(w)$.
We saw that a language L is called simply generated, if it is possible to determine the suffix $\mathsf{old}^L_{\prec_{\mathrm{lex}}}(w)$ and replace it by the suffix $\mathsf{new}^L_{\prec_{\mathrm{lex}}}(w)$ in a unique way for all $w \in L \setminus \{w^L_{\max}\}$ without any knowledge about $\mathsf{pre}^L_{\prec_{\mathrm{lex}}}(w)$. So, a language L is called simply generated, if it is sufficient to read the suffix to be changed only for all $w \in L \setminus \{w^L_{\max}\}$ in order to compute $\mathsf{next}^L_{\prec_{\mathrm{lex}}}(w)$.

Example 3:
Considering Example 1, with $w = [_2 [_2]_2]_2$ we find $\mathsf{next}_{\prec_{\mathsf{lex}}}^{D_4^{\mathcal{R}}}(w) = [_2 [_1]_1]_2$. We get $\mathsf{pre}_{\prec_{\mathsf{lex}}}^{D_4^{\mathcal{R}}}(w) = [_2$, $\mathsf{old}_{\prec_{\mathsf{lex}}}^{D_4^{\mathcal{R}}}(w) = [_2]_2]_2$ and $\mathsf{new}_{\prec_{\mathsf{lex}}}^{D_4^{\mathcal{R}}}(w) = [_1]_1]_2$. It is easy to check that the Dyck language $D_4^{\mathcal{R}}$ with $\mathcal{R} := \{([_1,]_1), ([_2,]_2)\}$ is simply generated.

In Section 2 we formalize the successor function; for that purpose we have to define some functions that give information on the relation $\mathcal{R}$. Note that the algorithm uses $w = w_0 \dots w_{2n-1} \in D_{2n}^{\mathcal{R}}$.
The first three functions depend on the relation $\mathcal{R}$ and a given Dyck word w.
The function $\mathsf{succ}_{\mathsf{pair}}^{\mathcal{R}}$:
The function $\mathsf{succ}_{\mathsf{pair}}^{\mathcal{R}}$ computes to a given pair of brackets $[_a]_b$ the next pair of brackets according to the lexicographical order $\prec_{\text{lex}}$ and the relation $\mathcal{R}$.

$$\mathsf{succ}_{\mathsf{pair}}^{\mathcal{R}}([_a]_b) := \begin{cases} [_c]_d & \text{if} \quad ([_c,]_d) \in \mathcal{R} \ \wedge \ [_a]_b \prec_{\text{lex}} [_c]_d \ \wedge \\ & \qquad \big(\forall (r_1, r_2) \in \mathcal{R} \setminus \{([_a,]_b), ([_c,]_d)\}\big) \\ & \qquad \big(r_1\, r_2 \prec_{\text{lex}} [_a]_b \ \vee \ [_c]_d \prec_{\text{lex}} r_1\, r_2\big) \\ \text{undefined} & \text{otherwise} \end{cases} .$$

Note that for all $(r_1, r_2) \in \mathcal{R} \setminus \left\{ w_{\max}^{D_{2n}^{\mathcal{R}}} \right\}$, the function $\mathsf{succ}_{\mathsf{pair}}^{\mathcal{R}}(r_1\, r_2)$ is always defined.
The function $\mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}$:
The function $\mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}$ computes to a given closing bracket $]_b$ in a Dyck word the next closing bracket according to the lexicographical order $\prec_{\text{lex}}$ and the relation $\mathcal{R}$. Here, $[_a$ denotes the unique corresponding opening bracket to $]_b$ in the Dyck word w. It can be determined as described in Remark 1.

$$\mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}(w, i) := \begin{cases}]_d & \text{if} \quad w \in D_{2n}^{\mathcal{R}} \ \wedge \ w_i =]_b \ \wedge \\ & \qquad [_a \text{ corresponds to }]_b \text{ in } w \ \wedge \\ & \qquad ([_a,]_d) \in \mathcal{R} \ \wedge \ [_a]_b \prec_{\text{lex}} [_a]_d \ \wedge \\ & \qquad \big(\forall (r_1, r_2) \in \mathcal{R} \setminus \{([_a,]_b), ([_a,]_d)\}\big) \\ & \qquad \big(r_1\, r_2 \prec_{\text{lex}} [_a]_b \ \vee \ [_a]_d \prec_{\text{lex}} r_1\, r_2\big) \\ \text{undefined} & \text{otherwise} \end{cases} .$$

Note that for all closing brackets not being maximal, $\mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}$ is always defined.
The function $\mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}$:
The function $\mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}$ computes to a given closing bracket $]_b$ in a Dyck word the minimal closing bracket that corresponds to $[_a$ according to the lexicographical order $\prec_{\text{lex}}$ and the relation $\mathcal{R}$. Again, $[_a$ denotes the unique corresponding opening bracket to $]_b$ in the Dyck word w.

$$\mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w, i) := \begin{cases}]_d & \text{if} \quad w \in D_{2n}^{\mathcal{R}} \ \wedge \ w_i =]_b \ \wedge \\ & \qquad [_a \text{ corresponds to }]_b \text{ in } w \ \wedge \\ & \qquad ([_a,]_d) \in \mathcal{R} \ \wedge \\ & \qquad \big(\forall ([_a, r) \in \mathcal{R} \setminus \{([_a,]_d)\}\big) \\ & \qquad \big([_a]_d \prec_{\text{lex}} [_a\, r\big) \\ \text{undefined} & \text{otherwise} \end{cases} .$$

Note that for all closing brackets, $\mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}$ is always defined. Further, the function $\mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}$ can be applied to $w_i \dots w_j \in T_]^{\star}$; we define

$$\mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w, i, j) := \\ \mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w, i)\, \mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w, i+1) \dots \mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w, j) \ .$$

If $w_i \dots w_j = \varepsilon$, we obtain $\mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w, i, j) = \varepsilon$.

Remark 2:
The functions $\mathsf{succ}^{\mathcal{R}}_{\mathsf{bracket}}$ and $\mathsf{min\_pred}^{\mathcal{R}}_{\mathsf{bracket}}$ need some more information, i. e. the corresponding opening bracket to the function's argument, which is a closing bracket. For some relations $\mathcal{R}$ it is necessary to read that opening bracket; for others it is not necessary, because the information needed can be determined by the relation $\mathcal{R}$. A condition will be given in the next section.

The following boolean functions depend on the relation $\mathcal{R}$ only.

<u>The function $\mathsf{no\_succ}^{\mathcal{R}}_{\mathsf{bracket}}$:</u>

The function $\mathsf{no\_succ}^{\mathcal{R}}_{\mathsf{bracket}}$ is true, if a given closing bracket $]_b$ is maximal with respect to all pairs of brackets it appears in.

$$\mathsf{no\_succ}^{\mathcal{R}}_{\mathsf{bracket}}(]_b) := \begin{cases} \text{true} & \text{if} \quad \big(\forall [_a \in T_{[}\big)\big(([_a,]_b) \in \mathcal{R} \\ & \qquad \rightsquigarrow \mathsf{succ}^{\mathcal{R}}_{\mathsf{pair}}([_a\,]_b) \notin \{[_a\,]_d \mid\,]_d \in T_{]}\}\big) \\ \text{false} & \text{otherwise} \end{cases} .$$

<u>The function $\mathsf{succ\_unique}^{\mathcal{R}}_{\mathsf{bracket}}$:</u>

The function $\mathsf{succ\_unique}^{\mathcal{R}}_{\mathsf{bracket}}$ is true, if the result of $\mathsf{succ}^{\mathcal{R}}_{\mathsf{bracket}}$ for a given closing bracket $]_b$ is the same – independent of the corresponding opening bracket to $]_b$.

$$\mathsf{succ\_unique}^{\mathcal{R}}_{\mathsf{bracket}}(]_b) := \begin{cases} \text{true} & \text{if} \quad \big(\forall [_{a_1}, [_{a_2} \in T_{[}\big)\big(([_{a_1},]_b), ([_{a_2},]_b) \in \mathcal{R} \\ & \qquad \rightsquigarrow \big(\big(\mathsf{succ}^{\mathcal{R}}_{\mathsf{pair}}([_{a_1}\,]_b) = [_{a_1}\,]_d \;\wedge \\ & \qquad\qquad \mathsf{succ}^{\mathcal{R}}_{\mathsf{pair}}([_{a_2}\,]_b) = [_{a_2}\,]_d \;\wedge \\ & \qquad\qquad]_d \in T_{]}\big) \;\vee \\ & \qquad\quad \big(\mathsf{succ}^{\mathcal{R}}_{\mathsf{pair}}([_{a_1}\,]_b) = \text{undefined} \;\wedge \\ & \qquad\qquad \mathsf{succ}^{\mathcal{R}}_{\mathsf{pair}}([_{a_2}\,]_b) = \text{undefined}\big)\big)\big) \\ \text{false} & \text{otherwise} \end{cases} .$$

<u>The function $\mathsf{min\_pred\_unique}^{\mathcal{R}}_{\mathsf{bracket}}$:</u>

The function $\mathsf{min\_pred\_unique}^{\mathcal{R}}_{\mathsf{bracket}}$ is true, if the result of $\mathsf{min\_pred}^{\mathcal{R}}_{\mathsf{bracket}}$ for a given closing bracket $]_b$ is the same – independent of the corresponding opening bracket to $]_b$.

$$\begin{aligned} &\mathsf{min\_pred\_unique}^{\mathcal{R}}_{\mathsf{bracket}}(]_b) \\ &:= \begin{cases} \text{true} & \text{if} \quad \big(\forall [_{a_1}, [_{a_2} \in T_{[}\big)\big(\exists\,]_d \in T_{]}\big)\big(([_{a_1},]_b), ([_{a_2},]_b) \in \mathcal{R} \\ & \qquad \rightsquigarrow \big(\big(\forall ([_{a_1},]_{c_1}) \in \mathcal{R} \setminus \{([_{a_1},]_d)\}\big)\big([_{a_1}\,]_d \prec_{\text{lex}} [_{a_1}\,]_{c_1}\big) \;\wedge \\ & \qquad\qquad \big(\forall ([_{a_2},]_{c_2}) \in \mathcal{R} \setminus \{([_{a_2},]_d)\}\big)\big([_{a_2}\,]_d \prec_{\text{lex}} [_{a_2}\,]_{c_2}\big)\big)\big) \\ \text{false} & \text{otherwise} \end{cases} . \end{aligned}$$

Example 4:
Let us revisit Example 2 with $\mathcal{R} := \{([_1,]_1), ([_1,]_2), ([_2,]_2)\}$. As the lexicographical order on the pairs of brackets is given by $[_2\,]_2 \prec_{\text{lex}} [_1\,]_1 \prec_{\text{lex}} [_1\,]_2$, we obtain:
$\mathsf{succ}^{\mathcal{R}}_{\mathsf{pair}}([_2\,]_2) = [_1\,]_1$, $\mathsf{succ}^{\mathcal{R}}_{\mathsf{pair}}([_1\,]_1) = [_1\,]_2$, $\mathsf{succ}^{\mathcal{R}}_{\mathsf{pair}}([_1\,]_2) = \text{undefined}$.
Now, we regard

$$w = w_0\, w_1\, w_2\, w_3\, w_4\, w_5 = [_1\,]_2\, [_1\, [_2\,]_2\,]_1 \in D^{\mathcal{R}}_6 \;.$$

Obviously,
$\mathsf{succ}^{\mathcal{R}}_{\mathsf{bracket}}(w, 1) = \text{undefined}$, $\mathsf{succ}^{\mathcal{R}}_{\mathsf{bracket}}(w, 4) = \text{undefined}$, $\mathsf{succ}^{\mathcal{R}}_{\mathsf{bracket}}(w, 5) =]_2$,
$\mathsf{min\_pred}^{\mathcal{R}}_{\mathsf{bracket}}(w, 1) =]_1$, $\mathsf{min\_pred}^{\mathcal{R}}_{\mathsf{bracket}}(w, 4, 5) =]_2\,]_1$.
The boolean functions immediately yield
$\mathsf{no\_succ}^{\mathcal{R}}_{\mathsf{bracket}}(]_1) = false$, $\mathsf{no\_succ}^{\mathcal{R}}_{\mathsf{bracket}}(]_2) = true$,
$\mathsf{succ\_unique}^{\mathcal{R}}_{\mathsf{bracket}}(]_1) = true$, $\mathsf{succ\_unique}^{\mathcal{R}}_{\mathsf{bracket}}(]_2) = true$,
$\mathsf{min\_pred\_unique}^{\mathcal{R}}_{\mathsf{bracket}}(]_1) = true$, $\mathsf{min\_pred\_unique}^{\mathcal{R}}_{\mathsf{bracket}}(]_2) = false$.

In Section 3 we formalize the condition on the relation $\mathcal{R}$, if a given language $D_{2n}^{\mathcal{R}}$ is simply generated or not. Further, we present an algorithm that decides, whether or not a relation $\mathcal{R}$ results in a simply generated Dyck language.

Definition 7:
Let $\mathsf{open}^{\mathcal{R}}(\,]_j) := \{[_i \mid ([_i,]_j) \in \mathcal{R}\}$ (resp. $\mathsf{close}^{\mathcal{R}}([_i) := \{\,]_j \mid ([_i,]_j) \in \mathcal{R}\}$) be the *set of all corresponding opening* (resp. *closing*) *brackets* to a given closing (resp. opening) bracket according to the relation $\mathcal{R}$. Further, let

$$\mathsf{successor}^{\mathcal{R}}(\,]_j,[_i) := \begin{cases}]_k & \text{if} \quad ([_i,]_k) \in \mathcal{R} \ \wedge\ [_i\,]_j \prec_{\text{lex}} [_i\,]_k \ \wedge \\ & \qquad (\forall\,]_l \in \mathsf{close}^{\mathcal{R}}([_i) \setminus \{\,]_j,]_k\}) \\ & \qquad\qquad ([_i\,]_l \prec_{\text{lex}} [_i\,]_j \ \vee\ [_i\,]_k \prec_{\text{lex}} [_i\,]_l) \\ \text{undefined} & \text{otherwise} \end{cases}$$

be the function that computes the same for a relation $\mathcal{R}$ as $\mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}$ does for a closing bracket of a Dyck word. The only difference is that we have to tell that function the corresponding closing bracket to the opening bracket, because it is not able to determine it as $\mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}$ is (, because $\mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}$ gets the whole word as parameter, not only a closing bracket). □

Note that $\mathsf{open}^{\mathcal{R}}$, $\mathsf{close}^{\mathcal{R}}$ and $\mathsf{successor}^{\mathcal{R}}$ depend on the relation $\mathcal{R}$ only and not on a Dyck word.

Definition 8:
The *matrix representation* of a relation $\mathcal{R}$ is given by

$$M^{\mathcal{R}} := (m_{i,j})_{1 \le i \le |T_[|, 1 \le j \le |T_]|} \quad \text{with} \quad m_{i,j} := \begin{cases} 1 & \text{if} \quad ([_i,]_j) \in \mathcal{R} \\ 0 & \text{if} \quad ([_i,]_j) \notin \mathcal{R} \end{cases}.$$

□

Example 5:
Considering the relation $\mathcal{R} := \{([_1,]_1),([_1,]_2),([_2,]_2)\}$ of Example 2, we immediately find $\mathsf{open}^{\mathcal{R}}(]_1) = \{[_1\}$, $\mathsf{open}^{\mathcal{R}}(]_2) = \{[_1,[_2\}$, $\mathsf{close}^{\mathcal{R}}([_1) = \{]_1,]_2\}$, $\mathsf{close}^{\mathcal{R}}([_2) = \{]_2\}$. The matrix representation of $\mathcal{R}$ is given by $M^{\mathcal{R}} = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$.

In Section 4 we analyze the length of the suffix to be changed in order to generate the next word according to the lexicographical order on the average. This average-case analysis is based on a general approach to the average length of the shortest suffix to be changed when generating words of a language lexicographically [Ke98].

2 Algorithm for the Generation of Dyck Words

In this section we will present an algorithm that generates all words $w \in D_{2n}^{\mathcal{R}}$ lexicographically. The following function $\mathsf{generate}$ starts with the generation of the lexicographically minimal word $w_{\min}^{D_{2n}^{\mathcal{R}}}$ and successively generates the next word $\mathsf{next}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w)$ of $w \in D_{2n}^{\mathcal{R}}$ according to the lexicographical order until the lexicographically maximal word $w_{\max}^{D_{2n}^{\mathcal{R}}}$ has been generated.

```
function generate ()
  begin
    w := w_min^{D_2n^R};
    while w ≠ undefined do
      w := next_{≺lex}^{D_2n^R}(w);
  end;
```

Algorithm 1: Function that generates all words of the language $D_{2n}^{\mathcal{R}}$ lexicographically.

Note that in the function generate the lexicographical generation is a transformation from one word to the next one according to the lexicographical order; it is called *successor*. So, each word $w \in D_{2n}^{\mathcal{R}} \setminus \left\{ w_{\min}^{D_{2n}^{\mathcal{R}}} \right\}$ depends on its *predecessor* only, there is no need of more information about the words previously generated.
In the following Lemma 1 we prove that $D_{2n}^{\mathcal{R}}$ can be splitted in four disjoint sets. In Lemma 2 we show and that each Dyck word $w = w_0 \dots w_{2n-1} \in D_{2n}^{\mathcal{R}}$ has a unique factorization.
Given a string s, we write $a \in s$ to express that a is a symbol occuring in s.
Lemma 1:
Let

$$\begin{aligned}
S_{2n}^{(1)} &:= \left\{ (p_{\max}^{\mathcal{R}})^n \right\} = \left\{ w_{\max}^{D_{2n}^{\mathcal{R}}} \right\} ,\\
S_{2n}^{(2)} &:= \{ x\, w_\alpha\, w_{\alpha+1} \dots w_\beta\, (p_{\max}^{\mathcal{R}})^i \mid \\
&\qquad\quad x \in T^{+},\ w_\alpha \in T_{]},\ w_\alpha \text{ is not maximal},\\
&\qquad\quad w_{\alpha+1} \dots w_\beta \in T_{]}^{\star},\ (\forall u \in w_{\alpha+1} \dots w_\beta)(u \text{ is maximal}),\\
&\qquad\quad 0 \le i \le n-1 \} \ \cap\ D_{2n}^{\mathcal{R}} ,\\
S_{2n}^{(3)} &:= \{ x\, w_\alpha\, w_{\alpha+1}\, w_{\alpha+2} \dots w_\beta\, (p_{\max}^{\mathcal{R}})^i \mid \\
&\qquad\quad x \in T^{\star},\ (w_\alpha, w_{\alpha+1}) \in \mathcal{R} \setminus \{\mathcal{R}_{\max}\},\ w_{\alpha+1} \text{ is maximal},\\
&\qquad\quad w_{\alpha+2} \dots w_\beta \in T_{]}^{\star}, (\forall u \in w_{\alpha+2} \dots w_\beta)(u \text{ is maximal}),\\
&\qquad\quad 0 \le i \le n-1 \} \ \cap\ D_{2n}^{\mathcal{R}} ,\\
S_{2n}^{(4)} &:= \{ x\, p_{\max}^{\mathcal{R}}\, w_\alpha \dots w_\beta\, (p_{\max}^{\mathcal{R}})^i \mid \\
&\qquad\quad x \in T^{+},\ w_\alpha \dots w_\beta \in T_{]}^{+},\ (\forall u \in w_\alpha \dots w_\beta)(u \text{ is maximal}),\\
&\qquad\quad 0 \le i \le n-2 \} \ \cap\ D_{2n}^{\mathcal{R}} .
\end{aligned}$$

Then the following equation holds:

$$D_{2n}^{\mathcal{R}} = S_{2n}^{(1)} \,\dot{\cup}\, S_{2n}^{(2)} \,\dot{\cup}\, S_{2n}^{(3)} \,\dot{\cup}\, S_{2n}^{(4)} .$$

Proof:
Obviously, $\left(\forall w \notin D_{2n}^{\mathcal{R}}\right)\left(\forall j \in [1:4]\right)\left(w \notin S_{2n}^{(j)}\right)$. Now, we have to prove that $\left(\forall w \in D_{2n}^{\mathcal{R}}\right)\left(\exists j \in [1:4]\right)\left(w \in S_{2n}^{(j)}\right)$. We assume that $\left(\exists w \in D_{2n}^{\mathcal{R}}\right)\left(\forall j \in [1:4]\right)\left(w \notin S_{2n}^{(j)}\right)$. Let $w = a\,(p_{\max}^{\mathcal{R}})^k$, $a \in T^{\star}$, $a \neq b\,p_{\max}^{\mathcal{R}}$, $b \in T^{\star}$. So, $(p_{\max}^{\mathcal{R}})^k$ is the suffix of w with maximal length consisting only of the lexicographically maximal pair of brackets, hence we get $0 \le k \le n$.
<u>Case 1:</u> $k = n$.
$\rightsquigarrow a = \varepsilon \rightsquigarrow w \in S_{2n}^{(1)}$.
<u>Case 2:</u> $k = n-1$.

$$\rightsquigarrow a = a_1\, a_2,\ (a_1, a_2) \in \mathcal{R} \rightsquigarrow \begin{cases} w \in S_{2n}^{(2)} & \text{if} \quad a_2 \text{ is not maximal} \\ w \in S_{2n}^{(3)} & \text{if} \quad a_2 \text{ is maximal} \end{cases} .$$

<u>Case 3:</u> $0 \le k \le n-2$.
$\rightsquigarrow a \in D^{\mathcal{R}}$, $\mathsf{length}(a) = 2n - 2k \ge 4$. Let $a = b\,c\,d$, $b \in T^{+}$, $c \in T$, c is not a maximal closing bracket, $d \in T_{]}^{\star}, (\forall v \in d)(v \text{ is maximal})$. So, d is the suffix of a with maximal length consisting only of maximal closing brackets. Now, we take a closer look at c.
<u>Case i:</u> $c \in T_{]}$.
$\rightsquigarrow c$ is not maximal $\rightsquigarrow w \in S_{2n}^{(2)}$.

Case ii: $c \in T_{[}$.

$\rightsquigarrow \mathsf{length}(d) \geq 1 \rightsquigarrow d = d_1\, d_2, \mathsf{length}(d_1) = 1$

$\rightsquigarrow \begin{cases} w \in S_{2n}^{(3)} & \text{if} \quad c\, d_1 \neq p_{\max}^{\mathcal{R}} \\ w \in S_{2n}^{(4)} & \text{if} \quad c\, d_1 = p_{\max}^{\mathcal{R}} \end{cases}$.

Altogether, $\big(\forall w \in D_{2n}^{\mathcal{R}}\big)\big(\exists j \in [1:4]\big)\big(w \in S_{2n}^{(j)}\big)$.

Now, we have to prove that the sets $S_{2n}^{(1)}$, $S_{2n}^{(2)}$, $S_{2n}^{(3)}$ and $S_{2n}^{(4)}$ are pairwise disjoint. In each of the following six cases we assume that $\big(\exists w \in D_{2n}^{\mathcal{R}}\big)\big(w \in S_{2n}^{(i)} \cap S_{2n}^{(j)}\big)$, $1 \leq i < j \leq 4$.

Note that the closing bracket of $p_{\max}^{\mathcal{R}}$, i. e. $]_{p_{\max}^{\mathcal{R}}}$, is maximal.

Case 1: $S_{2n}^{(1)} \cap S_{2n}^{(2)} \neq \emptyset$.

$\rightsquigarrow w_{\max}^{D_{2n}^{\mathcal{R}}} \in S_{2n}^{(2)}$. This is a contradiction.

Case 2: $S_{2n}^{(1)} \cap S_{2n}^{(3)} \neq \emptyset$.

$\rightsquigarrow w_{\max}^{D_{2n}^{\mathcal{R}}} \in S_{2n}^{(3)}$. This is a contradiction.

Case 3: $S_{2n}^{(1)} \cap S_{2n}^{(4)} \neq \emptyset$.

$\rightsquigarrow w_{\max}^{D_{2n}^{\mathcal{R}}} \in S_{2n}^{(4)}$. This is a contradiction.

Case 4: $S_{2n}^{(2)} \cap S_{2n}^{(3)} \neq \emptyset$.

We assume the existence of $w \in S_{2n}^{(2)} \cap S_{2n}^{(3)}$ with

$$w \;=\; x\, w_\alpha\, w_{\alpha+1} \ldots w_\beta\, (p_{\max}^{\mathcal{R}})^i \;=\; y\, v_\gamma\, v_{\gamma+1}\, v_{\gamma+2} \ldots v_\delta\, (p_{\max}^{\mathcal{R}})^j \;,$$

where

$x \in T^+$,
$w_\alpha \in T_{]}$, w_α is not maximal,
$w_{\alpha+1} \ldots w_\beta \in T_{]}^\star$, $(\forall u \in w_{\alpha+1} \ldots w_\beta)(u$ is maximal$)$,
$0 \leq i \leq n-1$,
$y \in T^\star$,
$(v_\gamma, v_{\gamma+1}) \in \mathcal{R} \setminus \{\mathcal{R}_{\max}\}$, $v_{\gamma+1}$ is maximal,
$v_{\gamma+2} \ldots v_\delta \in T_{]}^\star$, $(\forall u \in v_{\gamma+2} \ldots v_\delta)(u$ is maximal$)$,
$0 \leq j \leq n-1$.

As w_α is not maximal and $(v_\gamma, v_{\gamma+1}) \neq \mathcal{R}_{\max}$, we find $i = j$.

$\rightsquigarrow x\, w_\alpha \ldots w_\beta = y\, v_\gamma \ldots v_\delta$.

So, both words have a suffix consisting of maximal closing brackets.

$\rightsquigarrow w_{\alpha+1} \ldots w_\beta = v_{\gamma+1} \ldots v_\delta$

$\rightsquigarrow w_\alpha = v_\gamma$.

This is a contradiction, because $w_\alpha \in T_{]}$, $v_\gamma \in T_{[}$ and $T_{[} \cap T_{]} = \emptyset$.

Case 5: $S_{2n}^{(2)} \cap S_{2n}^{(4)} \neq \emptyset$.

We assume that $w \in S_{2n}^{(2)} \cap S_{2n}^{(4)}$ exists with

$$w \;=\; x\, w_\alpha\, w_{\alpha+1} \ldots w_\beta\, (p_{\max}^{\mathcal{R}})^i \;=\; y\, p_{\max}^{\mathcal{R}}\, v_\gamma \ldots v_\delta\, (p_{\max}^{\mathcal{R}})^j \;,$$

where

$x \in T^+$,
$w_\alpha \in T_{]}$, w_α is not maximal,
$w_{\alpha+1} \ldots w_\beta \in T_{]}^\star$, $(\forall u \in w_{\alpha+1} \ldots w_\beta)(u$ is maximal$)$,
$0 \leq i \leq n-1$,
$y \in T^+$,
$v_\gamma \ldots v_\delta \in T_{]}^+$, $(\forall u \in v_\gamma \ldots v_\delta)(u$ is maximal$)$,
$0 \leq j \leq n-2$.

As $w_\alpha \in T_{]}$, w_α is not maximal, $]_{p_{\max}^{\mathcal{R}}} \in T_{]}$ and $\mathsf{length}(v_\gamma \ldots v_\delta) \geq 1$, we find $i = j$.

$\rightsquigarrow x\, w_\alpha \ldots w_\beta = y\, p_{\max}^{\mathcal{R}}\, v_\gamma \ldots v_\delta$.

Since w_α is not maximal, we obtain $w_{\alpha+1} \ldots w_\beta =]_{p^{\mathcal{R}}_{\max}} v_\gamma \ldots v_\delta$.
$\rightsquigarrow w_\alpha = [_{p^{\mathcal{R}}_{\max}}$.
This is a contradiction, because $w_\alpha \in T_]$, $[_{p^{\mathcal{R}}_{\max}} \in T_[$ and $T_[\cap T_] = \emptyset$.

Case 6: $S^{(3)}_{2n} \cap S^{(4)}_{2n} \neq \emptyset$.
We assume the existence of $w \in S^{(3)}_{2n} \cap S^{(4)}_{2n}$, where

$$w \;=\; x\, w_\alpha\, w_{\alpha+1}\, w_{\alpha+2} \ldots w_\beta\, (p^{\mathcal{R}}_{\max})^i \;=\; y\, p^{\mathcal{R}}_{\max}\, v_\gamma \ldots v_\delta\, (p^{\mathcal{R}}_{\max})^j$$

with
$x \in T^\star$,
$(w_\alpha, w_{\alpha+1}) \in \mathcal{R} \setminus \{\mathcal{R}_{\max}\}$, $w_{\alpha+1}$ is maximal,
$w_{\alpha+2} \ldots w_\beta \in T_]^\star$, $(\forall u \in w_{\alpha+2} \ldots w_\beta)(u$ is maximal$)$,
$0 \leq i \leq n-1$,
$y \in T^+$,
$v_\gamma \ldots v_\delta \in T_]^+$, $(\forall u \in v_\gamma \ldots v_\delta)(u$ is maximal$)$,
$0 \leq j \leq n-2$.
By $w_\alpha, [_{p^{\mathcal{R}}_{\max}} \in T_[$, $w_{\alpha+1},]_{p^{\mathcal{R}}_{\max}} \in T_]$, $w_{\alpha+2} \ldots w_\beta \in T_]^\star$ and $v_\gamma \ldots v_\delta \in T_]^+$ we obtain $i = j$.
$\rightsquigarrow x\, w_\alpha\, w_{\alpha+1}\, w_{\alpha+2} \ldots w_\beta = y\, p^{\mathcal{R}}_{\max}\, v_\gamma \ldots v_\delta$.
As both words have a suffix consisting of closing brackets, we get $w_\alpha\, w_{\alpha+1} = p^{\mathcal{R}}_{\max}$.
This is a contradiction, because $(w_\alpha, w_{\alpha+1}) \in \mathcal{R} \setminus \{\mathcal{R}_{\max}\}$. ■

Remark 3:
If $n = 1$, we find $S^{(4)}_2 = \emptyset$.

Lemma 2:
Each Dyck word $w \in D^{\mathcal{R}}_{2n}$ has a unique factorization.

Proof:
As we have shown that $S^{(1)}_{2n}$, $S^{(2)}_{2n}$, $S^{(3)}_{2n}$ and $S^{(4)}_{2n}$ are pairwise disjoint, we only have to prove that for each word w there is only one factorization according to the set it belongs to.

$w \in S^{(1)}_{2n}$:
Trivial.

$w \in S^{(2)}_{2n}$:
We assume that two different factorizations of

$$w \;=\; x\, w_\alpha\, w_{\alpha+1} \ldots w_\beta\, (p^{\mathcal{R}}_{\max})^i \;=\; y\, v_\gamma\, v_{\gamma+1} \ldots v_\delta\, (p^{\mathcal{R}}_{\max})^j$$

exist with
$x, y \in T^+$,
$w_\alpha, v_\gamma \in T_]$, w_α, v_γ are not maximal,
$w_{\alpha+1} \ldots w_\beta, v_{\gamma+1} \ldots v_\delta \in T_]^\star$, $(\forall u \in w_{\alpha+1} \ldots w_\beta, v_{\gamma+1} \ldots v_\delta)(u$ is maximal$)$,
$0 \leq i, j \leq n-1$.
Assume without loss of generality $i < j \rightsquigarrow j - i > 0$.
$\rightsquigarrow x\, w_\alpha\, w_{\alpha+1} \ldots w_\beta = y\, v_\gamma\, v_{\gamma+1} \ldots v_\delta\, (p^{\mathcal{R}}_{\max})^{(j-i)}$
$\rightsquigarrow w_{\alpha+1} \ldots w_\beta = \varepsilon \vee (w_\alpha \in T_[\;\wedge\; w_{\alpha+1} \ldots w_\beta \in T_])$
$\rightsquigarrow w_\alpha \in T_]$ is maximal; this is a contradiction.
$\rightsquigarrow j - i = 0 \rightsquigarrow i = j$.
Now, assume without loss of generality $\mathsf{length}(w_{\alpha+1} \ldots w_\beta) < \mathsf{length}(v_{\gamma+1} \ldots v_\delta)$.
$\rightsquigarrow w_\alpha$ is a maximal closing bracket; this is a contradiction.
$\rightsquigarrow \mathsf{length}(w_{\alpha+1} \ldots w_\beta) = \mathsf{length}(v_{\gamma+1} \ldots v_\delta)$
$\rightsquigarrow w_{\alpha+1} \ldots w_\beta = v_{\gamma+1} \ldots v_\delta$.
$w_\alpha \neq v_\gamma$ is a contradiction, because $x\, w_\alpha = y\, v_\gamma$, $w_\alpha, v_\gamma \in T_]$.
$\rightsquigarrow w_\alpha = v_\gamma$
$\rightsquigarrow x = y$.

So, there is one factorization for every word $w \in S_{2n}^{(2)}$ only.

$\underline{w \in S_{2n}^{(3)}:}$

We assume the existence of two different factorizations of a word $w \in S_{2n}^{(3)}$ with

$$w \;=\; x\, w_\alpha\, w_{\alpha+1}\, w_{\alpha+2} \dots w_\beta\, (p_{\max}^{\mathcal{R}})^i \;=\; y\, v_\gamma\, v_{\gamma+1}\, v_{\gamma+2} \dots v_\delta\, (p_{\max}^{\mathcal{R}})^j \;,$$

where

$x, y \in T^\star$,
$(w_\alpha, w_{\alpha+1}), (v_\gamma, v_{\gamma+1}) \in \mathcal{R} \setminus \{\mathcal{R}_{\max}\}$, $w_{\alpha+1}, v_{\gamma+1}$ are maximal,
$w_{\alpha+2} \dots w_\beta,\, v_{\gamma+2} \dots v_\delta \in T_]^\star$, $(\forall u \in w_{\alpha+2} \dots w_\beta, v_{\gamma+2} \dots v_\delta)(u$ is maximal$)$,
$0 \le i, j \le n-1$.

Now, we assume without loss of generality $i < j \rightsquigarrow j - i > 0$.
$\rightsquigarrow x\, w_\alpha \dots w_\beta = y\, v_\gamma \dots v_\delta\, (p_{\max}^{\mathcal{R}})^{j-i}$
$\rightsquigarrow w_{\alpha+2} \dots w_\beta = \varepsilon \wedge (w_\alpha, w_{\alpha+1}) = \mathcal{R}_{\max}$.
That is a contradiction to $(w_\alpha, w_{\alpha+1}) \in \mathcal{R} \setminus \{\mathcal{R}_{\max}\}$.
$\rightsquigarrow j - i = 0 \rightsquigarrow i = j$.
Now, assume without loss of generality $\mathsf{length}(w_{\alpha+2} \dots w_\beta) < \mathsf{length}(v_{\gamma+2} \dots v_\delta)$.
$\rightsquigarrow w_\alpha = v_{\gamma+1} \in T_] \vee w_\alpha \in T_]$.
This is a contradiction to $w_\alpha \in T_[$, because $T_[\cap T_] = \emptyset$.
$\rightsquigarrow \mathsf{length}(w_{\alpha+2} \dots w_\beta) = \mathsf{length}(v_{\gamma+2} \dots v_\delta)$
$\rightsquigarrow w_{\alpha+2} \dots w_\beta = v_{\gamma+2} \dots v_\delta$.
$w_\alpha\, w_{\alpha+1} \neq v_\gamma\, v_{\gamma+1}$ is a contradiction, because $x\, w_\alpha\, w_{\alpha+1} = y\, v_\gamma\, v_{\gamma+1}$ and further $\mathsf{length}(w_\alpha\, w_{\alpha+1}) = \mathsf{length}(v_\gamma\, v_{\gamma+1}) = 2$.
$\rightsquigarrow w_\alpha\, w_{\alpha+1} = v_\gamma\, v_{\gamma+1}$
$\rightsquigarrow x = y$.
So, there is one factorization for every word $w \in S_{2n}^{(3)}$ only.

$\underline{w \in S_{2n}^{(4)}:}$

We assume that two different factorizations of

$$w \;=\; x\, p_{\max}^{\mathcal{R}}\, w_\alpha \dots w_\beta\, (p_{\max}^{\mathcal{R}})^i \;=\; y\, p_{\max}^{\mathcal{R}}\, v_\gamma \dots v_\delta\, (p_{\max}^{\mathcal{R}})^j$$

exist, where we have

$x, y \in T^+$,
$w_\alpha \dots w_\beta, v_\gamma \dots v_\delta \in T_]^+$, $(\forall u \in w_\alpha \dots w_\beta, v_\gamma \dots v_\delta)(u$ is maximal$)$,
$0 \le i, j \le n-2$.

Assume without loss of generality $i < j \rightsquigarrow j - i > 0$.
$\rightsquigarrow x\, p_{\max}^{\mathcal{R}}\, w_\alpha \dots w_\beta = y\, p_{\max}^{\mathcal{R}}\, v_\gamma \dots v_\delta\, (p_{\max}^{\mathcal{R}})^{j-i}$
$\rightsquigarrow w_\alpha \dots w_\beta = \varepsilon$.
That is a contradiction to $w_\alpha \dots w_\beta \in T_]^+$.
$\rightsquigarrow j - i = 0 \rightsquigarrow i = j$.
Now, assume without loss of generality $\mathsf{length}(w_\alpha \dots w_\beta) < \mathsf{length}(v_\gamma \dots v_\delta)$.
$\rightsquigarrow [_{p_{\max}^{\mathcal{R}}} \in T_]$.
This is a contradiction to $[_{p_{\max}^{\mathcal{R}}} \in T_[$, because $T_[\cap T_] = \emptyset$.
$\rightsquigarrow \mathsf{length}(w_\alpha \dots w_\beta) = \mathsf{length}(v_\gamma \dots v_\delta)$
$\rightsquigarrow w_\alpha \dots w_\beta = v_\gamma \dots v_\delta$.
Thus, we obtain $x = y$.
So, there is one factorization for every word $w \in S_{2n}^{(4)}$ only. ■

Now, we are able to formalize the successor function $\mathsf{next}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w)$ of a word $w \in D_{2n}^{\mathcal{R}}$ according to the lexicographical order. With Lemma 1 and Lemma 2 we can compute the successor for each of the sets $S_{2n}^{(j)}$, $1 \le j \le 4$, defined in Lemma 1.

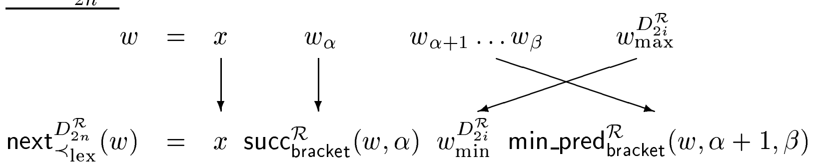
Theorem 1:

$$\mathsf{next}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w)$$

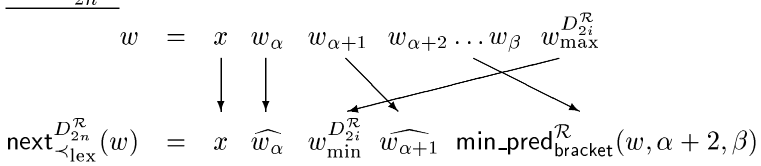
$$:= \begin{cases} \text{undefined} \quad \text{if} \quad w = (p_{\max}^{\mathcal{R}})^n \in S_{2n}^{(1)} \\ x\, \mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}(w,\alpha)\,([_{p_{\min}^{\mathcal{R}}})^i\,(]_{p_{\min}^{\mathcal{R}}})^i\, \mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w,\alpha+1,\beta) \\ \qquad \text{if} \quad w = x\, w_\alpha\, w_{\alpha+1} \dots w_\beta\, (p_{\max}^{\mathcal{R}})^i \in S_{2n}^{(2)}, \\ \qquad\quad x \in T^+, \\ \qquad\quad w_\alpha \in T_], \ w_\alpha \text{ is not maximal}, \\ \qquad\quad w_{\alpha+1} \dots w_\beta \in T_]^\star, \ (\forall u \in w_{\alpha+1} \dots w_\beta)(u \text{ is maximal}), \\ \qquad\quad 0 \le i \le n-1 \\ x\, \widehat{w_\alpha}\,([_{p_{\min}^{\mathcal{R}}})^i\,(]_{p_{\min}^{\mathcal{R}}})^i\, \widehat{w_{\alpha+1}}\, \mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w,\alpha+2,\beta) \\ \qquad \text{if} \quad w = x\, w_\alpha\, w_{\alpha+1}\, w_{\alpha+2} \dots w_\beta\, (p_{\max}^{\mathcal{R}})^i \in S_{2n}^{(3)}, \\ \qquad\quad x \in T^\star, \\ \qquad\quad (w_\alpha, w_{\alpha+1}) \in \mathcal{R} \setminus \{\mathcal{R}_{\max}\}, \ w_{\alpha+1} \text{ is maximal}, \\ \qquad\quad w_{\alpha+2} \dots w_\beta \in T_]^\star, \ (\forall u \in w_{\alpha+2} \dots w_\beta)(u \text{ is maximal}), \\ \qquad\quad 0 \le i \le n-1 \text{ and} \\ \qquad\quad \widehat{w_\alpha}\,\widehat{w_{\alpha+1}} = \mathsf{succ}_{\mathsf{pair}}^{\mathcal{R}}(w_\alpha\, w_{\alpha+1}), \ \widehat{w_\alpha} \in T_[, \ \widehat{w_{\alpha+1}} \in T_] \\ x\, \mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w,\alpha)\,([_{p_{\min}^{\mathcal{R}}})^{i+1}\,(]_{p_{\min}^{\mathcal{R}}})^{i+1}\, \mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w,\alpha+1,\beta) \\ \qquad \text{if} \quad w = x\, p_{\max}^{\mathcal{R}}\, w_\alpha\, w_{\alpha+1} \dots w_\beta\, (p_{\max}^{\mathcal{R}})^i \in S_{2n}^{(4)}, \\ \qquad\quad x \in T^+, \\ \qquad\quad w_\alpha \in T_], \ w_\alpha \text{ is maximal}, \\ \qquad\quad w_{\alpha+1} \dots w_\beta \in T_]^\star, \ (\forall u \in w_{\alpha+1} \dots w_\beta)(u \text{ is maximal}), \\ \qquad\quad 0 \le i \le n-2 \end{cases}.$$

Before we prove the theorem, let us have a closer look at the computation of the successor of a Dyck word $w \in D_{2n}^{\mathcal{R}} \setminus \left\{ w_{\max}^{D_{2n}^{\mathcal{R}}} \right\}$.

<u>$w \in S_{2n}^{(2)}$:</u>

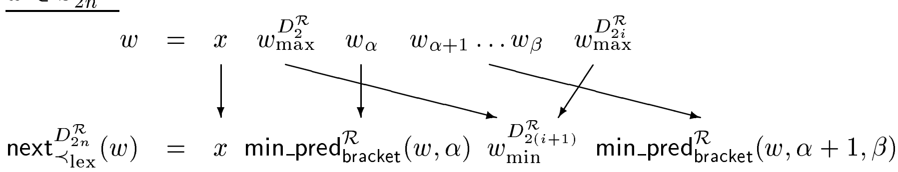


<u>$w \in S_{2n}^{(3)}$:</u>



Here, $\mathsf{succ}_{\mathsf{pair}}^{\mathcal{R}}(w_\alpha\ w_{\alpha+1}) = \widehat{w_\alpha}\ \widehat{w_{\alpha+1}}$, $\widehat{w_\alpha} \in T_[$, $\widehat{w_{\alpha+1}} \in T_]$.

<u>$w \in S_{2n}^{(4)}$:</u>



Proof:
We can prove this theorem by proving the correctness of the successor function for each case $w \in S_{2n}^{(j)}$, $1 \le j \le 4$, because every word $w \in D_{2n}^{\mathcal{R}}$ belongs to one of the sets $S_{2n}^{(j)}$, $1 \le j \le 4$, exactly.

$\underline{w \in S_{2n}^{(1)}:}$

Obviously, $w_{\max}^{D_{2n}^{\mathcal{R}}}$ has no successor.

$\underline{w \in S_{2n}^{(2)}:}$

Assume that $\big(\exists v \in D_{2n}^{\mathcal{R}}\big)\big(w \prec_{\text{lex}} v \prec_{\text{lex}} \mathsf{next}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w)\big) \rightsquigarrow v = x\,\tilde{v},\ x \in T^{+}$, with

$$w_\alpha\, w_{\alpha+1} \ldots w_\beta\, (p_{\max}^{\mathcal{R}})^i \quad \prec_{\text{lex}} \quad \tilde{v} \quad \prec_{\text{lex}} \quad \mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}(w,\alpha)\, ([_{p_{\min}^{\mathcal{R}}})^i\, (]_{p_{\min}^{\mathcal{R}}})^i\, \mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w,\alpha+1,\beta)\ ,$$

where

$w_\alpha \in T_{]}$, w_α is not maximal,
$w_{\alpha+1} \ldots w_\beta \in T_{]}^{\star}$, $(\forall u \in w_{\alpha+1} \ldots w_\beta)(u$ is maximal$)$,
$0 \le i \le n-1$.

Obviously, $w_\alpha <_{\text{lex}} \mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}(w,\alpha)$. Note that w_α and $\mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}(w,\alpha)$ are the only possibilities for the first symbol in $\tilde{v}$. Further, we know that every $v \in D_{2n}^{\mathcal{R}}$ has exactly n opening and n closing brackets.

Case 1: $\tilde{v} = w_\alpha\, \hat{v}$.

$\rightsquigarrow w_{\alpha+1} \ldots w_\beta\, (p_{\max}^{\mathcal{R}})^i \prec_{\text{lex}} \hat{v}$. This is a contradiction, because $\hat{v}$ must contain i opening brackets and $(\mathsf{length}(w_{\alpha+1} \ldots w_\beta) + i)$ closing brackets, but all closing brackets in $w_{\alpha+1} \ldots w_\beta$ are maximal, so none of them can be substituted by a lexicograpical larger one and $(p_{\max}^{\mathcal{R}})^i = w_{\max}^{D_{2i}^{\mathcal{R}}}$.
Additionally, $\big(\forall i \in [1 : |T_{[}|]\big)\big(\forall j \in [1 : |T_{]}|]\big)\big([_i <_{\text{lex}}]_j\big)$.
Thus, such $\hat{v}$ does not exist.

Case 2: $\tilde{v} = \mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}(w,\alpha)\, \hat{v}$.

$\rightsquigarrow \hat{v} \prec_{\text{lex}} ([_{p_{\min}^{\mathcal{R}}})^i\, (]_{p_{\min}^{\mathcal{R}}})^i\, \mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w,\alpha+1,\beta)$. This is a contradiction, because $\hat{v}$ must contain i opening brackets and $(\mathsf{length}(w_{\alpha+1} \ldots w_\beta) + i)$ closing brackets, but $([_{p_{\min}^{\mathcal{R}}})^i\, (]_{p_{\min}^{\mathcal{R}}})^i = w_{\min}^{D_{2i}^{\mathcal{R}}}$ and all closing brackets in $w_{\alpha+1} \ldots w_\beta$ are minimal.
Furthermore, $\big(\forall i \in [1 : |T_{[}|]\big)\big(\forall j \in [1 : |T_{]}|]\big)\big([_i <_{\text{lex}}]_j\big)$.
Hence, such $\hat{v}$ does not exist.

$\underline{w \in S_{2n}^{(3)}:}$

Assume that $\big(\exists v \in D_{2n}^{\mathcal{R}}\big)\big(w \prec_{\text{lex}} v \prec_{\text{lex}} \mathsf{next}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w)\big) \rightsquigarrow v = x\,\tilde{v},\ x \in T^{\star}$, with

$$w_\alpha\, w_{\alpha+1}\, w_{\alpha+2} \ldots w_\beta\, (p_{\max}^{\mathcal{R}})^i \quad \prec_{\text{lex}} \quad \tilde{v} \quad \prec_{\text{lex}} \quad \widehat{w_\alpha}\, ([_{p_{\min}^{\mathcal{R}}})^i\, (]_{p_{\min}^{\mathcal{R}}})^i \widehat{w_{\alpha+1}}\, \mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w,\alpha+2,\beta)\ ,$$

where

$(w_\alpha, w_{\alpha+1}) \in \mathcal{R} \setminus \{\mathcal{R}_{\max}\}$, $w_{\alpha+1}$ is maximal,
$w_{\alpha+2} \ldots w_\beta \in T_{]}^{\star}$, $(\forall u \in w_{\alpha+2} \ldots w_\beta)(u$ is maximal$)$,
$0 \le i \le n-1$,
$\widehat{w_\alpha}\, \widehat{w_{\alpha+1}} = \mathsf{succ}_{\mathsf{pair}}^{\mathcal{R}}(w_\alpha\, w_{\alpha+1})$.

Note that $w_\alpha <_{\text{lex}} \widehat{w_\alpha}$, because $w_{\alpha+1}$ is maximal and as there are no unused symbols in T, $(\forall y \in T_{[} \setminus \{w_\alpha, \widehat{w_\alpha}\})(y <_{\text{lex}} w_\alpha \vee \widehat{w_\alpha} <_{\text{lex}} y)$. Further, the first symbol in $\tilde{v}$ can be either w_α or $\widehat{w_\alpha}$. For the completion of a correct Dyck word, $\tilde{v}$ must contain $(i+1)$ opening and $(\mathsf{length}(w_{\alpha+2} \ldots w_\beta) + i + 1)$ closing brackets.

Case 1: $\tilde{v} = w_\alpha\, \hat{v}$.

$\rightsquigarrow w_{\alpha+1}\, w_{\alpha+2} \dots w_\beta\, (p^{\mathcal{R}}_{\max})^i \prec_{\text{lex}} \hat{v}$. This is a contradiction, because $w_{\alpha+1}$ and all brackets in $w_{\alpha+2} \dots w_\beta$ are maximal and $(p^{\mathcal{R}}_{\max})^i = w^{D^{\mathcal{R}}_{2i}}_{\max}$.

Furthermore, $\big(\forall i \in [1 : |T_{[}|]\big)\big(\forall j \in [1 : |T_{]}|]\big)\big([_i \prec_{\text{lex}}]_j\big)$.

Thus, such $\hat{v}$ does not exist.

Case 2: $\tilde{v} = \widehat{w_\alpha}\, \hat{v}$.

$\rightsquigarrow \hat{v} \prec_{\text{lex}} ([_{p^{\mathcal{R}}_{\min}})^i\, (]_{p^{\mathcal{R}}_{\min}})^i\, \widehat{w_{\alpha+1}}\, \mathsf{min\_pred}^{\mathcal{R}}_{\mathsf{bracket}}(w, \alpha{+}2, \beta)$. This is a contradiction, because $\widehat{w_{\alpha+1}}$ is minimal and $\mathsf{min\_pred}^{\mathcal{R}}_{\mathsf{bracket}}(w, \alpha + 2, \beta)$ contains only minimal opening brackets and further $([_{p^{\mathcal{R}}_{\min}})^i\, (]_{p^{\mathcal{R}}_{\min}})^i = w^{D^{\mathcal{R}}_{2i}}_{\min}$.

Additionally, $\big(\forall i \in [1 : |T_{[}|]\big)\big(\forall j \in [1 : |T_{]}|]\big)\big([_i \prec_{\text{lex}}]_j\big)$.

Hence, such $\hat{v}$ does not exist.

$w \in S^{(4)}_{2n}$:

Assume that $\big(\exists v \in D^{\mathcal{R}}_{2n}\big)\big(w \prec_{\text{lex}} v \prec_{\text{lex}} \mathsf{next}^{D^{\mathcal{R}}_{2n}}_{\prec_{\text{lex}}}(w)\big) \rightsquigarrow v = x\, \tilde{v},\ x \in T^\star$, with

$$p^{\mathcal{R}}_{\max}\, w_\alpha\, w_{\alpha+1} \dots w_\beta\, (p^{\mathcal{R}}_{\max})^i \quad \prec_{\text{lex}} \quad \tilde{v} \quad \prec_{\text{lex}}$$
$$\mathsf{min\_pred}^{\mathcal{R}}_{\mathsf{bracket}}(w, \alpha)\, ([_{p^{\mathcal{R}}_{\min}})^{i+1}\, (]_{p^{\mathcal{R}}_{\min}})^{i+1}\, \mathsf{min\_pred}^{\mathcal{R}}_{\mathsf{bracket}}(w, \alpha + 1, \beta)\ ,$$

where

$w_\alpha \in T_{]}$, w_α is maximal,

$w_{\alpha+1} \dots w_\beta \in T^\star_{]}$, $(\forall u \in w_{\alpha+1} \dots w_\beta)(u$ is maximal$)$,

$0 \le i \le n - 2$.

For the completion of a correct Dyck word, $\tilde{v}$ must contain $(i + 1)$ opening and $(\mathsf{length}(w_\alpha \dots w_\beta) + i + 1)$ closing brackets. Note that $[_{p^{\mathcal{R}}_{\max}}$ or $\mathsf{min\_pred}^{\mathcal{R}}_{\mathsf{bracket}}(w, \alpha)$ is the first symbol in $\tilde{v}$.

Case 1: $\tilde{v} = [_{p^{\mathcal{R}}_{\max}}\, \hat{v}$.

$\rightsquigarrow]_{p^{\mathcal{R}}_{\max}}\, w_\alpha\, w_{\alpha+1} \dots w_\beta\, (p^{\mathcal{R}}_{\max})^i \prec_{\text{lex}} \hat{v}$. This is a contradiction, because $]_{p^{\mathcal{R}}_{\max}}$, w_α and all brackets in $w_{\alpha+1} \dots w_\beta$ are maximal and $(p^{\mathcal{R}}_{\max})^i = w^{D^{\mathcal{R}}_{2i}}_{\max}$.

Furthermore, $\big(\forall i \in [1 : |T_{[}|]\big)\big(\forall j \in [1 : |T_{]}|]\big)\big([_i \prec_{\text{lex}}]_j\big)$.

Thus, such $\hat{v}$ does not exist.

Case 2: $\tilde{v} = \mathsf{min\_pred}^{\mathcal{R}}_{\mathsf{bracket}}(w, \alpha)\, \hat{v}$.

$\rightsquigarrow \hat{v} \prec_{\text{lex}} ([_{p^{\mathcal{R}}_{\min}})^{i+1}\, (]_{p^{\mathcal{R}}_{\min}})^{i+1}\, \mathsf{min\_pred}^{\mathcal{R}}_{\mathsf{bracket}}(w, \alpha + 1, \beta)$. This is a contradiction, because $([_{p^{\mathcal{R}}_{\min}})^{i+1}\, (]_{p^{\mathcal{R}}_{\min}})^{i+1} = w^{D^{\mathcal{R}}_{2(i+1)}}_{\min}$ is minimal and all brackets appearing in $\mathsf{min\_pred}^{\mathcal{R}}_{\mathsf{bracket}}(w, \alpha + 1, \beta)$ are minimal.

Further, $\big(\forall i \in [1 : |T_{[}|]\big)\big(\forall j \in [1 : |T_{]}|]\big)\big([_i \prec_{\text{lex}}]_j\big)$.

Hence, such $\hat{v}$ does not exist.

$\blacksquare$

Remark 4:

(i) When regarding the definition of $\mathsf{next}^{D^{\mathcal{R}}_{2n}}_{\prec_{\text{lex}}}(w)$ of $w \in D^{\mathcal{R}}_{2n}$, we notice that we need some information about the relation $\mathcal{R}$.

- Is a closing bracket maximal ?
- If a closing bracket is not maximal, what is its next closing bracket according to the relation $\mathcal{R}$ $\big(\widehat{=}\ \mathsf{succ}^{\mathcal{R}}_{\mathsf{bracket}}\big)$?
- If a closing bracket is maximal, what is its minimal closing bracket according to the relation $\mathcal{R}$ $\big(\widehat{=}\ \mathsf{min\_pred}^{\mathcal{R}}_{\mathsf{bracket}}\big)$?

In the next section we will focus on the condition for:

- This information can be obtained from the relation $\mathcal{R}$.

- This information can not be obtained from the relation $\mathcal{R}$, but it can be obtained by reading a part of the Dyck word, especially the corresponding opening bracket to that closing bracket.

(ii) if $n = 1$, the function $\mathsf{next}_{\prec_{\text{lex}}}^{D_2^{\mathcal{R}}}$ simplifies to:
$\mathsf{next}_{\prec_{\text{lex}}}^{D_2^{\mathcal{R}}}(w)$

$$:= \begin{cases} \text{undefined} & \text{if} \quad w = p_{\max}^{\mathcal{R}} \in S_2^{(1)} \\ w_0\, \mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}(w, 1) & \text{if} \quad w = w_0\, w_1 \in S_2^{(2)}, \\ & \quad\quad (w_0, w_1) \in \mathcal{R},\ w_1 \text{ is not maximal} \\ \mathsf{succ}_{\mathsf{pair}}^{\mathcal{R}}(w_0\, w_1) & \text{if} \quad w = w_0\, w_1 \in S_2^{(3)}, \\ & \quad\quad (w_0, w_1) \in \mathcal{R} \setminus \{\mathcal{R}_{\max}\},\ w_1 \text{ is maximal} \end{cases} .$$

As the lexicographical generation of all Dyck words of length 2 is equal to the lexicographical generation of all pairs of brackets, we immediately find: $\mathsf{next}_{\prec_{\text{lex}}}^{D_2^{\mathcal{R}}}(w) = \mathsf{succ}_{\mathsf{pair}}^{\mathcal{R}}(w)$.
Note that the function $\mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}$ is not needed, if $n = 1$, because $S_2^{(4)} = \emptyset$.

Now, we are able to formalize the function next. Regarding Theorem 1 we see that it has to read the Dyck word w from right to left. First, the algorithm reads all lexicographically maximal pairs of brackets $p_{\max}^{\mathcal{R}}$ at the end of w. Then, it has to read the string consisting of maximal closing brackets. Having read an opening bracket or a closing bracket not being maximal, the suffix to be changed is found and the successor can be generated.
Sometimes there are one or more closing brackets in that suffix, for which it is undecidable (in consideration of the suffix read only), whether the bracket is maximal or what the next or minimal closing bracket according to the order on the alphabet and the relation $\mathcal{R}$ is. If such a bracket is read, the function init is called. It reads to the left until the corresponding opening bracket is found (see Remark 1). The function init makes the information on this part of the Dyck word accessible to the functions $\mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}$ and $\mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}$ in the algorithm.
Now, we are able to formalize the algorithm that generates all words in $D_{2n}^{\mathcal{R}}$ lexicographically. Note that $w = w_0 \dots w_{2n-1} \in D_{2n}^{\mathcal{R}}$ in the algorithm.

dyck word function next(w: **dyck word**)
  begin
    $i := 2n - 1$
    $pairs := 0$
    $brackets := 0$
    $/\star\star\star\star\star$ find the suffix of w to be changed $\star\star\star\star\star/$
    $/\star$ read $w_{\max}^{D_{2pairs}^{\mathcal{R}}}$ $\star/$
    while $i \geq 1$ **and** $w_{i-1}\, w_i = p_{\max}^{\mathcal{R}}$ **do**
      begin
        $pairs := pairs + 1$
        $i := i - 2$
      end
    if $i < 1$ **then**
      begin
        $/\star$ successor for $w = w_{\max}^{D_{2n}^{\mathcal{R}}} \in S_{2n}^{(1)}$ does not exist $\star/$
        next := undefined
        return
      end

```
/⋆ read all maximal closing brackets on the left of w_max^{D^R_{2pairs}} ⋆/
while w_i ∈ T_] and no_succ^R_bracket(w_i) and min_pred_unique^R_bracket(w_i) do
   begin
      brackets := brackets + 1
      i := i − 1
   end
if w_i ∈ T_] and (not succ_unique^R_bracket(w_i)
      or (no_succ^R_bracket(w_i) and not min_pred_unique^R_bracket(w_i))) then
   begin
      init(w, i)
      while w_i ∈ T_] and w_i is maximal do
         begin
            brackets := brackets + 1
            i := i − 1
         end
   end
/⋆⋆⋆⋆⋆ change the suffix of w to generate its the successor ⋆⋆⋆⋆⋆/
if w_i ∈ T_] then
   begin
      /⋆ compute successor for w ∈ S^(2)_2n ⋆/
      w_i := succ^R_bracket(w, i)
      w_{2n−brackets} ... w_{2n−1} := min_pred^R_bracket(w,
                  2n − brackets − 2pairs, 2n − 1 − 2pairs)
      w_{i+1} ... w_{i+2pairs} := w_min^{D^R_{2pairs}}
   end
else
   begin
      if w_i w_{i+1} ≠ p^R_max then
         begin
            /⋆ compute successor for w ∈ S^(3)_2n ⋆/
            w_{2n+1−brackets} ... w_{2n−1} := min_pred^R_bracket(w,
                        2n + 1 − brackets − 2pairs, 2n − 1 − 2pairs)
            w_i w_{i+1} := succ^R_pair(w_i w_{i+1})
            w_{i+1+2pairs} := w_{i+1}
            w_{i+1} ... w_{i+2pairs} := w_min^{D^R_{2pairs}}
         end
      else
         begin
            /⋆ compute successor for w ∈ S^(4)_2n ⋆/
            w_i := min_pred^R_bracket(w, i + 2)
            w_{2n+1−brackets} ... w_{2n−1} := min_pred^R_bracket(w,
                        2n + 1 − brackets − 2pairs, 2n − 1 − 2pairs)
            w_{i+1} ... w_{i+2+2pairs} := w_min^{D^R_{2(pairs+1)}}
         end
   end
   next := w
end;
```

Algorithm 2: Successor function next for the lexicographical generation of Dyck words.

3 On the length of the suffixes read and changed

In the preceding section, we have noticed that for the generation of the successor of a word $w \in D_{2n}^{\mathcal{R}}$ according to the lexicographical order it might sometimes be necessary to read more than just the word's suffix to be changed (function init). In this section we will show that this neccessity depends on the relation $\mathcal{R}$.
Now, we formalize the condition for a simply generated Dyck language $D_{2n}^{\mathcal{R}}$. We will see that this condition depends on $\mathcal{R}$ only, if we distinguish between $n = 1$ and $n \geq 2$.

Theorem 2:
Case 1: $n = 1$.
$D_2^{\mathcal{R}}$ is simply generated $\iff$

$$\left(\forall\,]_j \in T_]\right)\left(\forall [_{i_1}, [_{i_2} \in \mathsf{open}^{\mathcal{R}}(]_j)\right)\left(\mathsf{successor}^{\mathcal{R}}(]_j, [_{i_1}) = \mathsf{successor}^{\mathcal{R}}(]_j, [_{i_2})\right) .$$

Case 2: $n \geq 2$.
$D_{2n}^{\mathcal{R}}$ is simply generated $\iff$

$$\left(\forall\,]_j \in T_]\right)\left(\forall [_{i_1}, [_{i_2} \in \mathsf{open}^{\mathcal{R}}(]_j)\right)\left(\mathsf{close}^{\mathcal{R}}([_{i_1}) = \mathsf{close}^{\mathcal{R}}([_{i_2})\right) .$$

Before proving the theorem, we take a look at some columns of two rows of the matrix representation $M^{\mathcal{R}}$ given in Figure 3 and Figure 4. We distinguish between two cases, which we will refer to in the proof.

$$\begin{pmatrix} & \vdots & & \vdots & \\ \cdots & m_{\kappa,\lambda} & 0 \cdots 0 & m_{\kappa,\lambda+\nu} & \cdots \\ & \vdots & & \vdots & \\ \cdots & m_{\kappa+\mu,\lambda} & \underbrace{0 \cdots 0}_{\geq 0} & m_{\kappa+\mu,\lambda+\nu} & \cdots \\ & \vdots & & \vdots & \end{pmatrix}$$

Figure 3: Matrix representation of a relation $\mathcal{R}$ (Case a).

In Figure 3, we have $1 \leq \kappa < \kappa + \mu \leq |T_[|$ and $1 \leq \lambda < \lambda + \nu \leq |T_]|$.
Case a: (Figure 3)
$\begin{pmatrix} m_{\kappa,\lambda} & m_{\kappa,\lambda+\nu} \\ m_{\kappa+\mu,\lambda} & m_{\kappa+\mu,\lambda+\nu} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ or $\begin{pmatrix} m_{\kappa,\lambda} & m_{\kappa,\lambda+\nu} \\ m_{\kappa+\mu,\lambda} & m_{\kappa+\mu,\lambda+\nu} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$.
Without loss of generality, we will refer to the left equation.

$$\begin{pmatrix} & \vdots & & \vdots & \\ 0 \cdots 0 & m_{\kappa,\lambda} & \cdots & m_{\kappa,\lambda+\nu} & 0 \cdots 0 \\ & \vdots & & \vdots & \\ \underbrace{0 \cdots 0}_{\geq 0} & m_{\kappa+\mu,\lambda} & \underbrace{\cdots}_{\geq 0} & m_{\kappa+\mu,\lambda+\nu} & \underbrace{0 \cdots 0}_{\geq 0} \\ & \vdots & & \vdots & \end{pmatrix}$$

Figure 4: Matrix representation of a relation $\mathcal{R}$ (Case b).

In Figure 4, we also have $1 \leq \kappa < \kappa + \mu \leq |T_[|$ and $1 \leq \lambda < \lambda + \nu \leq |T_]|$.
Case b: (Figure 4)
$\begin{pmatrix} m_{\kappa,\lambda} & m_{\kappa,\lambda+\nu} \\ m_{\kappa+\mu,\lambda} & m_{\kappa+\mu,\lambda+\nu} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ or $\begin{pmatrix} m_{\kappa,\lambda} & m_{\kappa,\lambda+\nu} \\ m_{\kappa+\mu,\lambda} & m_{\kappa+\mu,\lambda+\nu} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$.
Without loss of generality, we will refer to the left equation.

Proof:
Let us remember Remark 4. Generating the successor $\mathsf{next}_{\prec_{\mathrm{lex}}}^{D_{2n}^{\mathcal{R}}}(w)$ of $w \in D_{2n}^{\mathcal{R}}$, we need to decide, whether a given closing bracket $w_i =]_j \in T_]$ is maximal or not. If this is not the case, then we need to compute $\mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}(w,i)$; otherwise, we need to compute $\mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}(w,i)$ (but only if $n \geq 2$).

<u>Case 1:</u> $n = 1$.

<u>$\Rightarrow$:</u> We assume: $\big(\exists\,]_j \in T_]\big)\big(\exists [_{i_1}, [_{i_2} \in \mathsf{open}^{\mathcal{R}}(\,]_j)\big)$
$$\big(\mathsf{successor}^{\mathcal{R}}(\,]_j, [_{i_1}) \neq \mathsf{successor}^{\mathcal{R}}(\,]_j, [_{i_2})\big).$$
Obviously, this expression implies Case a.

<u>Case i:</u> $\{([_{i_1},]_j), ([_{i_1},]_a), ([_{i_2},]_j), ([_{i_2},]_b)\} \subseteq \mathcal{R},\ a \neq b \neq j,\ a \neq j\ \wedge$
$\mathsf{successor}^{\mathcal{R}}(\,]_j, [_{i_1}) = \,]_a \neq \,]_b = \mathsf{successor}^{\mathcal{R}}(\,]_j, [_{i_2})$.

Here, we have Case a with $\kappa = i_1$, $\kappa + \mu = i_2$, $\lambda = j$ and $\lambda + \nu = a$, where there is at least one 1 on the right side of $m_{i_2,a}$, of which the leftmost is at $m_{i_2,b} = 1$.

Consider
$$w = x\,]_j,\ x \in T_[\ \text{ with } \ \mathsf{next}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w) = x\,]_a$$
and
$$w' = y\,]_j,\ y \in T_[\ \text{ with } \ \mathsf{next}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w') = y\,]_b\ .$$

$\rightsquigarrow \mathsf{old}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w) = \alpha\, \mathsf{old}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w') = \,]_j$

$\rightsquigarrow \alpha = \varepsilon \wedge$
$\mathsf{new}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w) = \,]_a \neq \,]_b = \mathsf{new}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w')$

$\rightsquigarrow D_2^{\mathcal{R}}$ is not simply generated.

In this case we see that $D_2^{\mathcal{R}}$ can not be simply generated, if it is not possible to determine for arbitrary closing bracket the next one according to the relation $\mathcal{R}$ $\big(\widehat{=}\ \mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}\big)$ without knowing the corresponding opening bracket.

<u>Case ii:</u> $\{([_{i_1},]_j), ([_{i_1},]_a), ([_{i_2},]_j)\} \subseteq \mathcal{R},\ a \neq j\ \wedge$
$\mathsf{successor}^{\mathcal{R}}(\,]_j, [_{i_1}) = \,]_a \neq$ undefined $= \mathsf{successor}^{\mathcal{R}}(\,]_j, [_{i_2})$.

Here, we have Case a with κ, λ, μ and ν as in the previous case, but in this case are only 0's on the right side of $m_{i_2,a}$.

Again, consider
$$w = x\,]_j,\ x \in T_[\ \text{ with } \ \mathsf{next}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w) = x\,]_a$$
and
$$w' = y\,]_j,\ y \in T_[\ \text{ with } \ \mathsf{next}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w') \neq y\,]_c,\ ]_c \in T_]\ .$$

$\rightsquigarrow \mathsf{old}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w) = \,]_j\ \wedge$
$\mathsf{length}(\alpha\, \mathsf{old}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w')) > 1$

$\rightsquigarrow \mathsf{length}(\alpha) > 0$

$\rightsquigarrow \alpha \neq \varepsilon$

$\rightsquigarrow D_2^{\mathcal{R}}$ is not simply generated.

In that case we see that $D_2^{\mathcal{R}}$ can not be simply generated, if it is impossible to decide, whether any closing bracket is maximal or not according to the relation $\mathcal{R}$ without knowing the corresponding opening bracket.

<u>$\Leftarrow$:</u> We assume that $D_2^{\mathcal{R}}$ is not simply generated, i. e.
$$\big(\exists w, w' \in D_2^{\mathcal{R}}\big)\big(\mathsf{old}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w) = \alpha\, \mathsf{old}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w'), \alpha \in T^\star \rightsquigarrow (\alpha, \mathsf{new}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w)) \neq (\varepsilon, \mathsf{new}_{\prec_{\mathrm{lex}}}^{D_2^{\mathcal{R}}}(w'))\big).$$

Case i: $\alpha \neq \varepsilon$.

In this case a closing bracket, say $]_j$, is maximal in w and not maximal in w', so the suffix to be changed of w is longer than the one of w'. Note that this case corresponds to Case 1, "⇒", Case ii.

$\rightsquigarrow \big(\exists\,]_j,]_a \in T_]\big)\big(\exists [_{i_1}, [_{i_2} \in \mathsf{open}^{\mathcal{R}}(\,]_j)\big)$
$$\big(\mathsf{successor}^{\mathcal{R}}(\,]_j, [_{i_1}) = \,]_a \neq \text{undefined} = \mathsf{successor}^{\mathcal{R}}(\,]_j, [_{i_2})\big).$$

Case ii: $\alpha = \varepsilon \ \wedge \ \mathsf{new}_{\prec_{\text{lex}}}^{D_2^{\mathcal{R}}}(w) \neq \mathsf{new}_{\prec_{\text{lex}}}^{D_2^{\mathcal{R}}}(w')$.

Analogously to Case 1, "⇒", Case i, we find the next closing bracket to $]_j$ according to the relation $\mathcal{R}$ to be different for (different) corresponding opening brackets in w and w'.

$\rightsquigarrow \big(\exists\,]_j,]_a,]_b \in T_]\big)\big(\exists [_{i_1}, [_{i_2} \in \mathsf{open}^{\mathcal{R}}(\,]_j)\big)$
$$\big(\mathsf{successor}^{\mathcal{R}}(\,]_j, [_{i_1}) = \,]_a \neq \,]_b = \mathsf{successor}^{\mathcal{R}}(\,]_j, [_{i_2})\big).$$

Case 2: $n \geq 2$.

⇒ : We assume: $\big(\exists\,]_j \in T_]\big)\big(\exists [_{i_1}, [_{i_2} \in \mathsf{open}^{\mathcal{R}}(\,]_j)\big)\big(\mathsf{close}^{\mathcal{R}}([_{i_1}) \neq \mathsf{close}^{\mathcal{R}}([_{i_2})\big)$.

Evidently, this expression implies either Case a or Case b.

Case i: Analogously to Case 1, "⇒", Case i.

Case ii: Like in Case 1, "⇒", Case ii.

Case iii: $\{([_{i_1},]_a), ([_{i_1},]_j), ([_{i_2},]_b), ([_{i_2},]_j)\} \subseteq \mathcal{R}$, $a \neq b \ \wedge \ a \neq j \ \wedge$
$\big(\forall\,]_c \in \mathsf{close}^{\mathcal{R}}(\,[_{i_1}) \setminus \{\,]_a\}\big)\big(\,]_a <_{\text{lex}} \,]_c\big) \ \wedge$
$\big(\forall\,]_c \in \mathsf{close}^{\mathcal{R}}(\,[_{i_1}) \setminus \{\,]_j\}\big)\big(\,]_c <_{\text{lex}} \,]_j\big) \ \wedge$
$\big(\forall\,]_c \in \mathsf{close}^{\mathcal{R}}(\,[_{i_2}) \setminus \{\,]_b\}\big)\big(\,]_b <_{\text{lex}} \,]_c\big) \ \wedge$
$\big(\forall\,]_c \in \mathsf{close}^{\mathcal{R}}(\,[_{i_2}) \setminus \{\,]_j\}\big)\big(\,]_c <_{\text{lex}} \,]_j\big)$.

Here, we have Case b with $\kappa = i_1$, $\kappa + \mu = i_2$, $\lambda = a$ and $\lambda + \nu = j$, where there are only 0's on the left side of $m_{i_2,b}$.

Consider
$$w = x\, r_1\, r_2\,]_j,\ x \in T^+ \ \text{ with } \ \mathsf{next}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w) = x\, \mathsf{succ}_{\mathsf{pair}}^{\mathcal{R}}(r_1\, r_2)\,]_a$$
and
$$w' = y\, r_1\, r_2\,]_j,\ y \in T^+ \ \text{ with } \ \mathsf{next}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w') = y\, \mathsf{succ}_{\mathsf{pair}}^{\mathcal{R}}(r_1\, r_2)\,]_b\ ,$$
where $(r_1, r_2) \in \mathcal{R} \setminus \{\mathcal{R}_{\max}\}$, r_2 is maximal, $]_j$ is maximal.

$\rightsquigarrow \mathsf{old}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w) = \alpha\, \mathsf{old}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w') = r_1\, r_2\,]_j$

$\rightsquigarrow \alpha = \varepsilon \ \wedge$
$\mathsf{new}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w) = \mathsf{succ}_{\mathsf{pair}}^{\mathcal{R}}(r_1\, r_2)\,]_a \neq \mathsf{succ}_{\mathsf{pair}}^{\mathcal{R}}(r_1\, r_2)\,]_b = \mathsf{new}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w')$

$\rightsquigarrow D_{2n}^{\mathcal{R}}$ is not simply generated.

In this case we see that $D_{2n}^{\mathcal{R}}$ can not be simply generated, if it is not possible to determine for arbitrary closing bracket the minimal one according to the relation $\mathcal{R}$ $\big(\widehat{=}\ \mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}\big)$ without knowing the corresponding opening bracket.

⇐ : We assume that $D_{2n}^{\mathcal{R}}$ is not simply generated, i. e.

$\big(\exists w, w' \in D_{2n}^{\mathcal{R}}\big)\big(\mathsf{old}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w) = \alpha\, \mathsf{old}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w'), \alpha \in T^\star \rightsquigarrow$
$$(\alpha, \mathsf{new}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w)) \neq (\varepsilon, \mathsf{new}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w'))\big).$$

Case i: $\alpha \neq \varepsilon$.

This case corresponds to Case 2, "⇒", Case ii.

It is like in Case 1, "⇐", Case i.

$\rightsquigarrow \mathsf{close}^{\mathcal{R}}([_{i_1}) \neq \mathsf{close}^{\mathcal{R}}([_{i_2})$.

Case ii: $\alpha = \varepsilon \ \wedge \ \mathsf{new}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w) \neq \mathsf{new}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w')$.

This yields one of the following cases:

- This case corresponds to Case 2, "$\Rightarrow$", Case i.
  It is the same as in Case 1, "$\Leftarrow$", Case ii.
  $\rightsquigarrow \mathsf{close}^{\mathcal{R}}([_{i_1}) \neq \mathsf{close}^{\mathcal{R}}([_{i_2})$.
- This case corresponds to Case 2, "$\Rightarrow$", Case iii.
  We obtain $]_a$ for w and $]_b$ for w' (or vice versa) with $a \neq b$ and $a \neq j$ as minimal closing bracket to $]_j$ according to the relation $\mathcal{R}$.
  $\rightsquigarrow \mathsf{close}^{\mathcal{R}}([_{i_1}) \neq \mathsf{close}^{\mathcal{R}}([_{i_2})$.

So, in each case we have found a contradiction. ■

Remark 5:

(i) From Theorem 2 follows immediately, that a Dyck language $D_{2n}^{\mathcal{R}}$ is simply generated for $n \geq 1$, if $\left|\mathsf{open}^{\mathcal{R}}(]_j)\right| = 1$, $1 \leq j \leq |T_]|$.

(ii) In order to compute $\mathsf{next}_{\prec_{\mathrm{lex}}}^{D_{2n}^{\mathcal{R}}}(w)$ of $w \in D_{2n}^{\mathcal{R}}$ we have to read a suffix of w. If $D_{2n}^{\mathcal{R}}$ is simply generated, we have to change exactly the brackets we need to read in order to generate the successor of $w \in D_{2n}^{\mathcal{R}}$. In that case, the information required by the functions $\mathsf{succ}_{\mathsf{bracket}}^{\mathcal{R}}$ and $\mathsf{min\_pred}_{\mathsf{bracket}}^{\mathcal{R}}$ and for the decision, whether a closing bracket is maximal, can be directly deduced from the relation $\mathcal{R}$. This is in contrast to a not simply generated language, whereas - for at least one word - a part of the common prefix of the Dyck word and its successor has to be read.

(iii) The familiar Dyck languages with the relations

- $\mathcal{R} = \{([_1,]_1)\}$, i. e. with one pair of brackets,
- $\mathcal{R} = \{([_1,]_1), \ldots, ([_t,]_t)\}$, i. e. with t pairs of brackets, $t \geq 1$, where every opening (resp. closing) bracket corresponds to one closing (resp. opening) bracket only and
- $\mathcal{R} = \{([_1,]_1), \ldots, ([_1,]_r), ([_2,]_1), \ldots, ([_2,]_r), \ldots, ([_l,]_1), \ldots, ([_l,]_r)\}$, i. e. with $l\,r$ pairs of brackets, $l, r \geq 1$, where each opening (resp. closing) bracket corresponds to each closing (resp. opening) bracket

are simply generated.

(iv) Let

$$\widetilde{M} := \begin{pmatrix} A^{(1)} & & & & \\ & A^{(2)} & & 0 & \\ & & \ddots & & \\ & 0 & & A^{(k-1)} & \\ & & & & A^{(k)} \end{pmatrix}$$

be the *normal form* of a matrix M. Here, $A^{(l)}$, $1 \leq l \leq k$, are submatrices.

- If $n = 1$, the form of $A^{(l)}$ is given as follows. The first row consists of 1's only. The following rows have the form $\underbrace{0 \cdots 0}_{\geq 0}\underbrace{1 \cdots 1}_{\geq 1}$, where the number of 1's is monotonic decreasing from the first to the last row. For example, the matrices

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

have that property.

- In the case $n \geq 2$, each entry in $A^{(l)}$ is equal to 1.

The matrix representation $M^{\mathcal{R}}$ of the Dyck language $D_{2n}^{\mathcal{R}}$ with relation $\mathcal{R}$ can be transformed to its normal-form $\widetilde{M}^{\mathcal{R}}$ by permutating rows and columns, iff $D_{2n}^{\mathcal{R}}$ is simply generated.

Example 6:
$D_{2n}^{\mathcal{R}}$ with relation $\mathcal{R} = \{([_1,]_1), ([_2,]_2)\}$ of Example 1 is simply generated for $n \geq 1$, as $|\mathsf{open}^{\mathcal{R}}(]_1)| = 1$ and $|\mathsf{open}^{\mathcal{R}}(]_2)| = 1$.
For the language $D_{2n}^{\mathcal{R}}$ with relation $\mathcal{R} = \{([_1,]_1), ([_1,]_2), ([_2,]_2)\}$ of Example 2 we find $\mathsf{open}^{\mathcal{R}}(]_1) = \{[_1\}$ and $\mathsf{open}^{\mathcal{R}}(]_2) = \{[_1, [_2\}$. As $\mathsf{successor}^{\mathcal{R}}(]_2, [_1) =$ undefined $= \mathsf{successor}^{\mathcal{R}}(]_2, [_2)$, $D_{2n}^{\mathcal{R}}$ is simply generated for $n = 1$. Since $\mathsf{close}^{\mathcal{R}}([_1) = \{]_1,]_2\} \neq \{]_2\} = \mathsf{close}^{\mathcal{R}}([_2)$, $D_{2n}^{\mathcal{R}}$ is not simply generated for $n \geq 2$.
Now, one can think about on how to decide algorithmicly, whether $D_{2n}^{\mathcal{R}}$ is simply generated or not. As we have seen, the property of being simply generated depends on the relation $\mathcal{R}$ only; we have to distinguish between $n = 1$ and $n \geq 2$. Let us have a look at an algorithm that decides, whether $D_{2n}^{\mathcal{R}}$ is simply generated or not. We discuss the algorithm for $n = 1$, an algorithm for $n \geq 2$ is similar.
The following algorithms checks, if the language $D_2^{\mathcal{R}}$ is simply generated.

```
boolean function simply_generated ()
  begin
    simply_generated := true
    /* check for all closing brackets ]_j, ... */
    for j := 1 to |T_]| do
      begin
        /* ... if successor^R(]_j, [_i) is the same for all corresponding [_i */
        k := 0
        for i := 1 to |T_[| do
          if m_{i,j} = 1 and k = 0 then
            k := search_next(i, j)
          else if m_{i,j} = 1 and k ≠ search_next(i, j) then
            begin
              simply_generated := false
              return
            end
      end
  end;
```

Algorithm 3: Function that checks, if the language $D_2^{\mathcal{R}}$ is simply generated.

The function simply_generated uses another function that computes $\mathsf{successor}^{\mathcal{R}}(]_j, [_i)$ for any $([_i,]_j) \in \mathcal{R}$ by looking up the column of the next 1 on the right side of the entry for $([_i,]_j)$ in $M^{\mathcal{R}}$ in the same row. If such 1 does not exist, $|T_]| + 1$ will be returned, which means $\mathsf{successor}^{\mathcal{R}}(]_j, [_i) =$ undefined. The function search_next is defined as follows.

```
integer function search_next (i, j : integer)
  begin
    search_next := j + 1
    while search_next ≤ |T_]| and m_{i,search_next} = 0 do
      search_next := search_next + 1
  end;
```

Algorithm 4: Function that looks up the column of the next 1 on the right of an entry in $M^{\mathcal{R}}$.

Note that each entry in the matrix $M^{\mathcal{R}}$ is regarded twice at most. So, it can be checked in $\mathcal{O}\big(\left|T_{[}\right|\left|T_{]}\right|\big)$, if the language $D_2^{\mathcal{R}}$ is simply generated. The same fact holds for the case $n \geq 2$. Hence, the amount to decide, whether the language is simply generated, is constant with respect to n, i. e. to the number of pairs of brackets in the words of the language.

4 Analysis of the Algorithm

In this section we analyze the length of the suffix to be changed in order to compute the successor $\mathsf{next}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w)$ of a Dyck word $w \in D_{2n}^{\mathcal{R}}$. Remember that for every word the suffix to be changed is equal to the suffix to be read, if $D_{2n}^{\mathcal{R}}$ is simply generated. If $D_{2n}^{\mathcal{R}}$ is not simply generated, then there is at least one Dyck word $w \in D_{2n}^{\mathcal{R}} \setminus \left\{w_{\max}^{D_{2n}^{\mathcal{R}}}\right\}$, for which an algorithm has to inspect the common prefix $\mathsf{pre}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w)$ of w and $\mathsf{next}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}(w)$. So, the length of the suffix to be read is greater than the length of the suffix to be changed.
Let $X_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$ be the random variable that describes the length of the suffix to be changed. Obviously, the function $\mathsf{generate}$ generates the Dyck words in $D_{2n}^{\mathcal{R}}$ with respect to the lexicographical order $\prec_{\text{lex}}$. Thus, every word is generated exactly once. Further, the function $\mathsf{next}_{\prec_{\text{lex}}}^{D_{2n}^{\mathcal{R}}}$ reads the words from right to left. Under these conditions, the s-th moments, $s \geq 1$, about the origin of the random variable $X_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$ are given by [Ke98]:

$$\mathbb{E}[X_{\mathsf{sufl}}^s(D_{2n}^{\mathcal{R}})] := 1 + \left|D_{2n}^{\mathcal{R}}\right|^{-1} \sum_{k=1}^{2n-1} \left[(k+1)^s - k^s\right] \left|\mathsf{INIT}_{2n-k}(D_{2n}^{\mathcal{R}})\right| . \qquad (1)$$

Here, $\mathsf{INIT}_k(D_{2n}^{\mathcal{R}}) := \mathsf{INIT}(D_{2n}^{\mathcal{R}}) \cap T^k$ denotes the *set of all prefixes of length k appearing in words belonging to $D_{2n}^{\mathcal{R}}$*; thereby, the *set of all prefixes appearing in words belonging to $D_{2n}^{\mathcal{R}}$* is defined by $\mathsf{INIT}(D_{2n}^{\mathcal{R}}) := \left\{u \in T^\star \mid (\exists v \in T^\star)\left(uv \in D_{2n}^{\mathcal{R}}\right)\right\}$.
Now, our interest is $\left|\mathsf{INIT}_k(D_{2n}^{\mathcal{R}})\right|$, $1 \leq k < 2n$. For that purpose we consider the well-known one-to-one-correspondence between the Dyck language and paths on the lattice given in Figure 5.

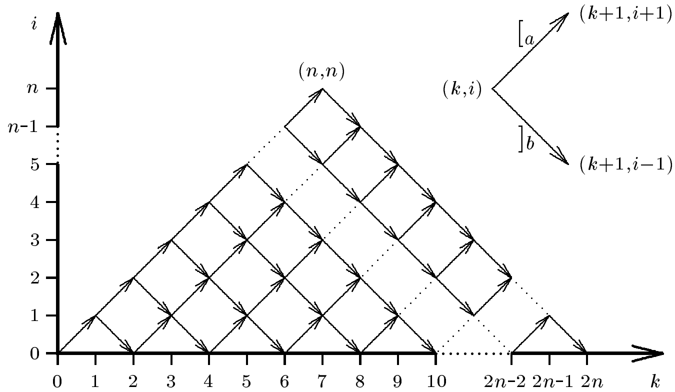


Figure 5: One-to-one correspondence between Dyck words of length $2n$ and the paths from $(0,0)$ to $(2n,0)$.

Each Dyck word of length $2n$ corresponds to a path from $(0,0)$ to $(2n,0)$. A segment $\nearrow$ (resp. $\searrow$) is labelled by $[_a$ (resp. $]_b$), where $[_a \in T_{[}$ (resp. $]_b \in T_{]}$). The Dyck word results from the concatenation of all labels on a path from $(0,0)$ to $(2n,0)$.

With the number of paths from $(0,0)$ to (k,i) in the lattice given in Figure 5 given by

$$\mathsf{p}(k,i) = \binom{k}{\frac{1}{2}(k-i)} - \binom{k}{\frac{1}{2}(k-i)-1} \tag{2}$$

[Ke96] we have

$$\left|\mathsf{INIT}_k(D_{2n}^{\mathcal{R}})\right| = \sum_{\substack{0\le i\le \min\{k,2n-k\}\\ k+i\,\equiv\,0 \bmod 2}} |\mathcal{R}|^{\frac{1}{2}(k-i)}\ \left|T_{[}\right|^{i}\ \mathsf{p}(k,i)\ . \tag{3}$$

This formula can be found in [Ke96], too.
Distinguishing between the terms for even k and those for odd k in (3), we immediately obtain by inserting (2):

$$\begin{aligned}&\left|\mathsf{INIT}_k(D_{2n}^{\mathcal{R}})\right|\\ &= \sum_{i=0}^{\min\{\lfloor\frac{k}{2}\rfloor, n-\lfloor\frac{k+1}{2}\rfloor\}} |\mathcal{R}|^{\lfloor\frac{k}{2}\rfloor-i}\ \left|T_{[}\right|^{2i+k-2\lfloor\frac{k}{2}\rfloor} \left[\binom{k}{\lfloor\frac{k}{2}\rfloor-i} - \binom{k}{\lfloor\frac{k}{2}\rfloor-i-1}\right].\end{aligned} \tag{4}$$

By (1), $\left|D_{2n}^{\mathcal{R}}\right| = \left|\mathsf{INIT}_{2n}(D_{2n}^{\mathcal{R}})\right|$ and (4), we get:

$$\begin{aligned}&\mathbb{E}[X_{\mathsf{sufl}}^{s}(D_{2n}^{\mathcal{R}})]\\ &= \left|D_{2n}^{\mathcal{R}}\right|^{-1} \sum_{k=0}^{2n-1} [(k+1)^s - k^s] \sum_{i=0}^{\min\{\lfloor\frac{2n-k}{2}\rfloor, n-\lfloor\frac{2n-k+1}{2}\rfloor\}} |\mathcal{R}|^{\lfloor\frac{2n-k}{2}\rfloor-i}\\ &\quad\times \left|T_{[}\right|^{2i+2n-k-2\lfloor\frac{2n-k}{2}\rfloor} \left[\binom{2n-k}{\lfloor\frac{2n-k}{2}\rfloor-i} - \binom{2n-k}{\lfloor\frac{2n-k}{2}\rfloor-i-1}\right].\end{aligned}$$

Splitting the first sum into two parts and letting the index of the second sum go to infinity, we obtain with $\left|D_{2n}^{\mathcal{R}}\right| = \frac{1}{n+1}\binom{2n}{n}|\mathcal{R}|^n$ after rearranging the sums and applying some simplifications:

$$\mathbb{E}[X_{\mathsf{sufl}}^{s}(D_{2n}^{\mathcal{R}})] = \frac{n+1}{\binom{2n}{n}} \left[F^{(1)}_{s,|\mathcal{R}|,|T_{[}|}(n) + F^{(2)}_{s,|\mathcal{R}|,|T_{[}|}(n) - F^{(3)}_{s,|\mathcal{R}|,|T_{[}|}(n) - F^{(4)}_{s,|\mathcal{R}|}(n)\right] ,$$

where

$$\begin{aligned}F^{(1)}_{s,a,b}(n) &:= \sum_{i\ge 0} a^{-i}\, b^{2i} \sum_{k=0}^{n} [(2k+1)^s - (2k)^s]\, a^{-k}\\ &\qquad\times\left[\binom{2n-2k}{n-k-i} - \binom{2n-2k}{n-k-i-1}\right] ,\\ F^{(2)}_{s,a,b}(n) &:= \sum_{i\ge 0} a^{-i-1}\, b^{2i+1} \sum_{k=0}^{n} [(2k+2)^s - (2k+1)^s]\, a^{-k}\\ &\qquad\times\left[\binom{2n-2k-1}{n-k-i-1} - \binom{2n-2k-1}{n-k-i-2}\right] ,\\ F^{(3)}_{s,a,b}(n) &:= \sum_{i\ge 0} a^{-i-1}\, b^{2i+2} \sum_{k=0}^{n} [(k+1)^s - k^s]\, a^{-k}\, b^{k}\\ &\qquad\times\left[\binom{2n-k}{n-k-i-1} - \binom{2n-k}{n-k-i-2}\right] ,\\ F^{(4)}_{s,a}(n) &:= \frac{(2n+1)^s - (2n)^s}{a^n} .\end{aligned}$$

In order to gain an asymptotic for $n \to \infty$ for $\mathbb{E}[X^s_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n})]$, we need to study the asymptotic behaviour of $F^{(1)}_{s,a,b}(n)$, $F^{(2)}_{s,a,b}(n)$, $F^{(3)}_{s,a,b}(n)$ and $F^{(4)}_{s,a}(n)$. For that purpose, we consider the generating functions for these functions.

Lemma 3:
Let $s, a, b \in \mathbb{N}$ and $u := \sqrt{1-4z}$. The generating functions of $F^{(1)}_{s,a,b}(n)$, $F^{(2)}_{s,a,b}(n)$, $F^{(3)}_{s,a,b}(n)$ and $F^{(4)}_{s,a}(n)$ are given by:

$$G^{(1)}_{s,a,b}(z) := \sum_{n\geq 0} F^{(1)}_{s,a,b}(n)\, z^n = \frac{2}{(1+u)\left(1 - \frac{b^2(1-u)^2}{4az}\right)} \times \left(\frac{1}{1-\frac{z}{a}} + \sum_{j=1}^{s-1} \binom{s}{j} 2^j\, \mathsf{A}_j\left(\frac{z}{a}\right) \frac{1}{\left(1-\frac{z}{a}\right)^{j+1}} \right) ,$$

$$G^{(2)}_{s,a,b}(z) := \sum_{n\geq 0} F^{(2)}_{s,a,b}(n)\, z^n = \frac{b(1-u)}{a(1+u)\left(1 - \frac{b^2(1-u)^2}{4az}\right)} \times \left(\frac{(-1)^{s+1}}{1-\frac{z}{a}} + \frac{a}{z}\sum_{j=1}^{s-1} \binom{s}{j} 2^j\, (-1)^{s-j+1}\, \mathsf{A}_j\left(\frac{z}{a}\right) \frac{1}{\left(1-\frac{z}{a}\right)^{j+1}} \right) ,$$

$$G^{(3)}_{s,a,b}(z) := \sum_{n\geq 0} F^{(3)}_{s,a,b}(n)\, z^n = \frac{4b}{(1+u)^2\left(1 - \frac{b^2(1-u)}{a(1+u)}\right)} \times \mathsf{A}_s\left(\frac{b(1-u)}{2a}\right) \frac{1}{\left(1-\frac{b(1-u)}{2a}\right)^s} ,$$

$$G^{(4)}_{s,a}(z) := \sum_{n\geq 0} F^{(4)}_{s,a}(n)\, z^n = \frac{1}{1-\frac{z}{a}} + \sum_{j=1}^{s-1} \binom{s}{j} 2^j\, \mathsf{A}_j\left(\frac{z}{a}\right) \frac{1}{\left(1-\frac{z}{a}\right)^{j+1}} .$$

Here, $\mathsf{A}_l(x)$ denotes the l-th Eulerian Polynomial [Ke84, p. 214] with

$$\mathsf{A}_l(x) = (1-x)^{l+1} \sum_{m\geq 0} m^l x^m = x \sum_{i=1}^{l} i! \left\{ {l \atop i} \right\} (x-1)^{l-i} ,$$

where $\left\{ {n \atop k} \right\}$ stands for the Stirling number of the second kind [GKP94, p. 258].

Proof:
In order to find the generating function for $F^{(1)}_{s,a,b}(n)$, we first consider the convolution of two functions.

$$\left(\sum_{\mu\geq 0} [(2\mu+1)^s - (2\mu)^s]\, a^{-\mu}\, z^\mu \right) \left(\sum_{\lambda\geq 0} \left[\binom{2\lambda}{\lambda-i} - \binom{2\lambda}{\lambda-i-1} \right] z^\lambda \right)$$
$$= \sum_{\kappa\geq 0} z^\kappa \sum_{\nu=0}^{\kappa} [(2\nu+1)^s - (2\nu)^s]\, a^{-\nu} \left[\binom{2\kappa-2\nu}{\kappa-\nu-i} - \binom{2\kappa-2\nu}{\kappa-\nu-i-1} \right] . \quad (5)$$

With $u = \sqrt{1-4z} \rightsquigarrow 4z = (1+u)(1-u)$ and the identity

$$\sum_{m\geq 0} \binom{2m+\alpha}{m} z^m = \frac{1}{\sqrt{1-4z}} \left(\frac{1-\sqrt{1-4z}}{2z} \right)^\alpha = \frac{1}{u} \left(\frac{1-u}{2z} \right)^\alpha , \alpha \in \mathbb{N}_0 , \quad (6)$$

[GKP94, p. 203] a straightforward computation leads to

$$\sum_{\lambda\geq 0} \left[\binom{2\lambda}{\lambda-i} - \binom{2\lambda}{\lambda-i-1} \right] z^\lambda = \frac{2}{1+u} \left(\frac{(1-u)^2}{4z} \right)^i . \quad (7)$$

In consideration of (5) and (7) we obtain

$$\begin{aligned} G^{(1)}_{s,a,b}(z) &= \sum_{i\geq 0} a^{-i}\, b^{2i} \left(\sum_{\lambda \geq 0} \left[\binom{2\lambda}{\lambda - i} - \binom{2\lambda}{\lambda - i - 1} \right] z^{\lambda} \right) \\ &\quad \times \left(\sum_{\mu \geq 0} [(2\mu+1)^s - (2\mu)^s]\, a^{-\mu}\, z^{\mu} \right) \\ &= \frac{2}{(1+u)\left(1 - \frac{b^2(1-u)^2}{4az}\right)} \left(\sum_{\mu\geq 0} [(2\mu+1)^s - (2\mu)^s]\, a^{-\mu}\, z^{\mu} \right) \qquad (8) \end{aligned}$$

by the expansion of the geometric series. Now, let us have a closer look at the sum in (8). Splitting off the first term and applying the binomial theorem to $(2\mu+1)^s$ and using again the expansion of the geometric series yields

$$\begin{aligned} \sum_{\mu\geq 0} [(2\mu+1)^s - (2\mu)^s]\, a^{-\mu}\, z^{\mu} &= \sum_{\mu \geq 0} \left(\frac{z}{a}\right)^{\mu} + \sum_{j=1}^{s-1} \binom{s}{j} 2^j \sum_{\mu\geq 0} \mu^j \left(\frac{z}{a}\right)^{\mu} \\ &= \frac{1}{1-\frac{z}{a}} + \sum_{j=1}^{s-1} \binom{s}{j} 2^j\, \mathsf{A}_j \left(\frac{z}{a}\right) \frac{1}{\left(1-\frac{z}{a}\right)^{j+1}} \,. \qquad (9) \end{aligned}$$

Inserting the expression (9) in (8) results in the generating function $G^{(1)}_{s,a,b}(z)$ for the numbers $F^{(1)}_{s,a,b}(n)$ stated in the lemma. $G^{(2)}_{s,a,b}(z)$ can be computed analogously. For the generating function $G^{(3)}_{s,a,b}(z)$ we consider the function

$$H_{s,c,d}(z) := \sum_{n\geq 0} z^n \sum_{k=0}^{n} [(k+1)^s - k^s] \binom{2n-k}{n+c} d^k \ , \ c \in \mathbb{N}_0 \, , \ d > 0 \ .$$

A rearrangement of the terms of $H_{s,c,d}(z)$ and the application of the identity $\binom{n}{k} = \binom{n}{n-k}$, $n,k \in \mathbb{N}_0$, results in

$$H_{s,c,d}(z) = \sum_{k\geq 0} [(k+1)^s - k^s]\, d^k\, z^{k+c} \sum_{n\geq 0} z^n \binom{2n+k+2c}{n}$$

and further with (6) we get

$$H_{s,c,d}(z) = \frac{1}{u} \left(\frac{1-u}{1+u}\right)^c \sum_{k\geq 0} [(k+1)^s - k^s] \left(\frac{d(1-u)}{2}\right)^k \ .$$

Moreover, a simple computation shows that

$$\sum_{k\geq 0} [(k+1)^s - k^s]\, y^k = \mathsf{A}_s(y)\, \frac{1}{y(1-y)^s}$$

holds; applying this identity, we obtain:

$$H_{s,c,d}(z) = \frac{1}{u}\left(\frac{1-u}{1+u}\right)^c \mathsf{A}_s \left(\frac{d(1-u)}{2}\right) \frac{1}{\frac{d(1-u)}{2}\left(1 - \frac{d(1-u)}{2}\right)^s} \ . \qquad (10)$$

Now, by (10), we get after simplifications

$$\begin{aligned}
G^{(3)}_{s,a,b}(z) &= \sum_{n\geq 0} z^n \sum_{i\geq 0} \left(\frac{b^2}{a}\right)^{i+1} \\
&\quad \times \sum_{k=0}^{n} [(k+1)^s - k^s] \left(\frac{b}{a}\right)^k \left[\binom{2n-k}{n+i+1} - \binom{2n-k}{n+i+2}\right] \\
&= \frac{4b}{(1+u)^2\left(1-\frac{b^2(1-u)}{a(1+u)}\right)} \mathsf{A}_s\left(\frac{b(1-u)}{2a}\right) \frac{1}{\left(1-\frac{b(1-u)}{2a}\right)^s} .
\end{aligned}$$

$G^{(4)}_{s,a}(z)$ can be calculated similar to $G^{(1)}_{s,a,b}(z)$. ∎

With this lemma, we are able to formalize
Theorem 3:
Let all $w \in D^{\mathcal{R}}_{2n}$ be equally likely. The s-th moments, $s \geq 1$, about the origin of the random variable $X_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n})$ are constant. They are given by:

$$\mathbb{E}[X^s_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n})] \sim C_{s,|\mathcal{R}|,|T_{[}|},\ n \to \infty, \text{ where}$$

$$\begin{aligned}
&C_{s,|\mathcal{R}|,|T_{[}|} \\
&:= \frac{|\mathcal{R}|}{\left(|\mathcal{R}| - |T_{[}|^2\right)^2} \left[\sum_{j=1}^{s-1} \binom{s}{j} 2^j \mathsf{A}_j\left(\frac{1}{4|\mathcal{R}|}\right) \frac{|\mathcal{R}| + |T_{[}|^2 + 4|\mathcal{R}||T_{[}|(-1)^{s-j+1}}{\left(1-\frac{1}{4|\mathcal{R}|}\right)^{j+1}} \right. \\
&\quad - \frac{2^s|\mathcal{R}|^{s-1}|T_{[}|}{(2|\mathcal{R}| - |T_{[}|)^{s+1}} \left[2|\mathcal{R}|\left(4|\mathcal{R}|^2 + (s-2)|\mathcal{R}||T_{[}| - s|T_{[}|^3\right) \mathsf{A}_s\left(\frac{|T_{[}|}{2|\mathcal{R}|}\right) \right. \\
&\quad \left. \left. + |T_{[}|\left(2|\mathcal{R}| - |T_{[}|\right)\left(|\mathcal{R}| - |T_{[}|^2\right) \mathsf{A}'_s\left(\frac{|T_{[}|}{2|\mathcal{R}|}\right) \right] + \frac{4|\mathcal{R}|\left(|\mathcal{R}| + |T_{[}|^2 - |T_{[}|(-1)^s\right)}{4|\mathcal{R}| - 1} \right].
\end{aligned}$$

Proof:
Using the results of the preceding lemma, we get the following formula after a simple computation.

$$\begin{aligned}
&\mathbb{E}[X^s_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n})] \\
&= \frac{n+1}{\binom{2n}{n}} [z^n] \left\{ G^{(1)}_{s,|\mathcal{R}|,|T_{[}|}(z) + G^{(2)}_{s,|\mathcal{R}|,|T_{[}|}(z) - G^{(3)}_{s,|\mathcal{R}|,|T_{[}|}(z) - G^{(4)}_{s,|\mathcal{R}|}(z) \right\} \\
&= \frac{n+1}{\binom{2n}{n}} [z^n] \left\{ \frac{1}{(1+u)\left(1-\frac{|T_{[}|^2(1-u)}{|\mathcal{R}|(1+u)}\right)} \left[\left(2 - (1+u)\left(1 - \frac{|T_{[}|^2(1-u)}{|\mathcal{R}|(1+u)}\right)\right) \right. \right. \\
&\quad \times \left(\frac{1}{1-\frac{z}{|\mathcal{R}|}} + \sum_{j=1}^{s-1} \binom{s}{j} 2^j \mathsf{A}_j\left(\frac{z}{|\mathcal{R}|}\right) \frac{1}{\left(1-\frac{z}{|\mathcal{R}|}\right)^{j+1}} \right) + (1-u)\frac{|T_{[}|}{|\mathcal{R}|} \\
&\quad \times \left(\frac{(-1)^{s+1}}{1-\frac{z}{|\mathcal{R}|}} + \frac{|\mathcal{R}|}{z} \sum_{j=1}^{s-1} \binom{s}{j} 2^j (-1)^{s-j+1} \mathsf{A}_j\left(\frac{z}{|\mathcal{R}|}\right) \frac{1}{\left(1-\frac{z}{|\mathcal{R}|}\right)^{j+1}} \right)
\end{aligned}$$

$$- \frac{4|T_{[}|}{(1+u)} \mathsf{A}_s \left(\frac{|T_{[}|(1-u)}{2|\mathcal{R}|}\right) \frac{1}{\left(1-\frac{|T_{[}|(1-u)}{2|\mathcal{R}|}\right)^s} \Bigg]\Bigg\} .$$

Now, we are looking for the singularity of smallest modulus that is not equal to 0. We find possible singularities at $z = 0$, $z = \frac{1}{4}$, $z = |\mathcal{R}|$, $z = \frac{|\mathcal{R}||T_{[}|^2}{(|\mathcal{R}|+|T_{[}|^2)^2}$ and $z = \frac{|\mathcal{R}|(|T_{[}|-|\mathcal{R}|)}{|T_{[}|^2}$. We do not have to take $z = 0$ and $z = |\mathcal{R}| > \frac{1}{4}$ into account. As an expansion of $G^{(1)}_{s,|\mathcal{R}|,|T_{[}|}(z) + G^{(2)}_{s,|\mathcal{R}|,|T_{[}|}(z) - G^{(3)}_{s,|\mathcal{R}|,|T_{[}|}(z) - G^{(4)}_{s,|\mathcal{R}|}(z)$ around $z = \frac{|\mathcal{R}||T_{[}|^2}{(|\mathcal{R}|+|T_{[}|^2)^2}$ or $z = \frac{|\mathcal{R}|(|T_{[}|-|\mathcal{R}|)}{|T_{[}|^2}$ results in a taylor series, i. e. neither $z = \frac{|\mathcal{R}||T_{[}|^2}{(|\mathcal{R}|+|T_{[}|^2)^2}$ nor $z = \frac{|\mathcal{R}|(|T_{[}|-|\mathcal{R}|)}{|T_{[}|^2}$ is a singularity, the singularity nearest to the origin is given by $z = \frac{1}{4}$. Expanding $G^{(1)}_{s,|\mathcal{R}|,|T_{[}|}(z) + G^{(2)}_{s,|\mathcal{R}|,|T_{[}|}(z) - G^{(3)}_{s,|\mathcal{R}|,|T_{[}|}(z) - G^{(4)}_{s,|\mathcal{R}|}(z)$ around $y := 1 - 4z$ yields:

$$\begin{aligned} & G^{(1)}_{s,|\mathcal{R}|,|T_{[}|}(z) + G^{(2)}_{s,|\mathcal{R}|,|T_{[}|}(z) - G^{(3)}_{s,|\mathcal{R}|,|T_{[}|}(z) - G^{(4)}_{s,|\mathcal{R}|}(z) \\ & \quad = \quad C'_{s,|\mathcal{R}|,|T_{[}|} - 2\, C_{s,|\mathcal{R}|,|T_{[}|} \sqrt{y} + \mathcal{O}(y) \ , \end{aligned}$$

where $C_{s,|\mathcal{R}|,|T_{[}|}$ is given in the theorem and $C'_{s,|\mathcal{R}|,|T_{[}|} = \mathcal{O}(1)$ is another constant. An application of Darboux's method [GK82] immediately gives the asymptotic behaviour of the s-th moments of the random variable $X_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n})$ stated in the theorem with $\frac{1}{n+1}\binom{2n}{n} \sim \frac{4^n}{\sqrt{\pi} n^{\frac{3}{2}}}$, $n \to \infty$, obtained by Stirling's formula [GKP94, p. 454]. ■

Corollary 1:
Considering $X_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n})$, the mean value $\mu_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n}) = \mathbb{E}[X^1_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n})]$ and the variance $\sigma^2_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n}) = \mathbb{E}[X^2_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n})] - \mathbb{E}[X^1_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n})]^2$ for $n \to \infty$ are given by:

$$\begin{aligned} \mu_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n}) &\sim \frac{16|\mathcal{R}|^3}{\big(4|\mathcal{R}|-1\big)\big(2|\mathcal{R}|-|T_{[}|\big)^2} \ , \\ \sigma^2_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n}) &\sim \frac{16|\mathcal{R}|^3\big(16|\mathcal{R}|^2|T_{[}| + 12|\mathcal{R}|^2 - 12|\mathcal{R}||T_{[}|^2 - 20|\mathcal{R}||T_{[}| + 7|T_{[}|^2\big)}{\big(4|\mathcal{R}|-1\big)^2\big(2|\mathcal{R}|-|T_{[}|\big)^4} \ . \end{aligned}$$

■

Remark 6:
Now, we focus on $\mu_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n})$. According to Theorem 3, we have $\mu_{\mathsf{sufl}}(D^{\mathcal{R}}_{2n}) \sim C_{1,|\mathcal{R}|,|T_{[}|}$, $n \to \infty$. A moment's reflection shows that

$$\begin{aligned} C_{1,1,1} &= \frac{16}{3} \ , \\ C_{1,|\mathcal{R}|,1} &= \frac{16|\mathcal{R}|^3}{\big(4|\mathcal{R}|-1\big)\big(2|\mathcal{R}|-1\big)^2} = 1 + \mathcal{O}\left(\frac{1}{|\mathcal{R}|}\right) \ , \ |\mathcal{R}| \to \infty \ , \\ C_{1,|\mathcal{R}|,|\mathcal{R}|} &= \frac{16|\mathcal{R}|}{\big(4|\mathcal{R}|-1\big)} = 4 + \mathcal{O}\left(\frac{1}{|\mathcal{R}|}\right) \ , \ |\mathcal{R}| \to \infty \end{aligned}$$

hold. We further find with $1 \le |T_{[}| \le |\mathcal{R}|$:

$$C_{1,|\mathcal{R}|,|T_{[}|} > C_{1,|\mathcal{R}|+1,|T_{[}|} \ , \tag{11}$$
$$C_{1,|\mathcal{R}|,|\mathcal{R}|} > C_{1,|\mathcal{R}|+1,|\mathcal{R}|+1} \ , \tag{12}$$
$$C_{1,|\mathcal{R}|,|T_{[}|} < C_{1,|\mathcal{R}|,|T_{[}|+1} \ . \tag{13}$$

Altogether, we obtain the following values for $C_{1,|\mathcal{R}|,|T_{[}|}$ given in Table 1. Here, ↑, ↖ and → stand for strictly increasing sequences. Obviously, ↑, ↖ and → correspond to (11), (12) and (13), respectively.

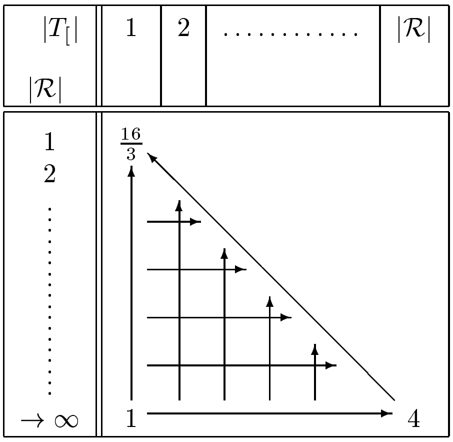


Table 1: Values for $C_{1,|\mathcal{R}|,|T_{[}|}$, $1 \leq |T_{[}| \leq |\mathcal{R}|, |\mathcal{R}| \to \infty$.

Now, we take a look at two plots of $C_{1,|\mathcal{R}|,|T_{[}|}$ for $1 \leq |T_{[}| \leq |\mathcal{R}| \leq 10$ from two different points of view, analytically continued to $\mathbb{R}$.

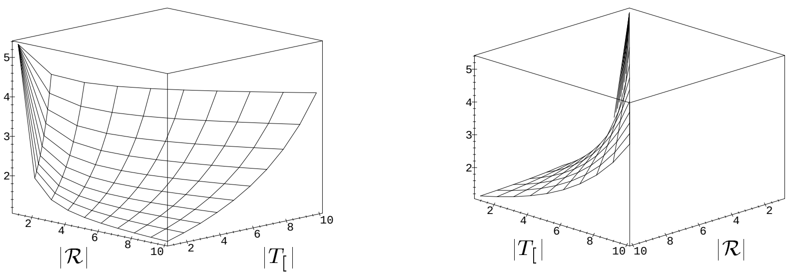


Figure 6: Two views of $C_{1,|\mathcal{R}|,|T_{[}|}$ for $1 \leq |T_{[}| \leq |\mathcal{R}| \leq 10$.

By these plots, we get a better idea of the behaviour of $\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$ for different combinations of the two parameters $|\mathcal{R}|$ and $|T_{[}|$. In the left plot of Figure 6, we see clearly the inequations (11) and (12), in the right plot we recognize (13).

Remark 7:

In [Li96] the lexicographical generation of the *Dyck language with t types of brackets* D^t [Ha78, p. 313] was analyzed. The results are presented in [Ke98]. The formal definition of D^t is as follows:

Let $t \in \mathbb{N}$ and $T := \{[_1,]_1, \ldots, [_t,]_t\}$ be the alphabet with the linear ordering $[_t <_{\text{lex}} \ldots <_{\text{lex}} [_1 <_{\text{lex}}]_1 <_{\text{lex}} \ldots <_{\text{lex}}]_t$. The Dyck language D^t is the smallest subset of $T^\star$ satisfying (i) and (ii):

(i) $\varepsilon \in D^t$,

(ii) $u, v \in D^t \leadsto [_1\, u\,]_1\, v \in D^t \wedge \ldots \wedge [_t\, u\,]_t\, v \in D^t$.

The Dyck language with n pairs of balanced brackets and t types of brackets is given by $D_{2n}^t := D^t \cap T^{2n}$. Note that $D_{2n}^t = D_{2n}^{\mathcal{R}}$ with the relation $\mathcal{R} = \{([_1,]_1), \ldots, ([_t,]_t)\}$. Obviously, D_{2n}^t is simply generated (see Remark 5).

The mean value and the variance of the random variable $X_{\mathsf{sufl}}(D_{2n}^t)$ describing the number of symbols to be changed while generating the successor of a word was found to be

$$\mu_{\mathsf{sufl}}(D_{2n}^{t}) \sim \frac{16t}{4t-1},\ n\to\infty \text{ and } \sigma_{\mathsf{sufl}}^{2}(D_{2n}^{t}) \sim \frac{16t}{4t-1},\ n\to\infty\ .$$

For $\mathcal{R} = \{([_1,]_1),\dots,([_t,]_t)\}$, we obtain $|\mathcal{R}| = \left|T_{[}\right| = t$ and so by Corollary 1 $\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}}) \sim \frac{16t}{4t-1}$, $n\to\infty$ and $\sigma_{\mathsf{sufl}}^{2}(D_{2n}^{\mathcal{R}}) \sim \frac{16t}{4t-1}$, $n\to\infty$, evidently.

In the following two tables (Table 2 and Table 3) we give some exact and asymptotical values for $\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$ and $\sigma_{\mathsf{sufl}}^{2}(D_{2n}^{\mathcal{R}})$ for various relations $\mathcal{R}$.

$\lvert\mathcal{R}\rvert$	n	$\lvert T_{[}\rvert$	1	2	3	4	5
1	10		4.91176				
	100	$\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$	5.28127				
	$\to\infty$		5.33333				
	10		3.73227				
	100	$\sigma_{\mathsf{sufl}}^{2}(D_{2n}^{\mathcal{R}})$	5.08069				
	$\to\infty$		5.33333				
2	10		2.04358	4.08490			
	100	$\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$	2.03273	4.51384			
	$\to\infty$		2.03175	4.57143			
	10		1.79522	2.65130			
	100	$\sigma_{\mathsf{sufl}}^{2}(D_{2n}^{\mathcal{R}})$	1.77458	4.29137			
	$\to\infty$		1.77375	4.57143			
3	10		1.58401	2.39785	3.86900		
	100	$\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$	1.57212	2.44870	4.30531		
	$\to\infty$		1.57091	2.45455	4.36364		
	10		0.97310	1.94598	2.40665		
	100	$\sigma_{\mathsf{sufl}}^{2}(D_{2n}^{\mathcal{R}})$	0.93482	2.20012	4.07994		
	$\to\infty$		0.93112	2.23140	4.36364		
4	10		1.40361	1.88379	2.61626	3.76953	
	100	$\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$	1.39417	1.89503	2.71870	4.20811	
	$\to\infty$		1.39320	1.89630	2.73067	4.26667	
	10		0.65391	1.26514	2.02532	2.29851	
	100	$\sigma_{\mathsf{sufl}}^{2}(D_{2n}^{\mathcal{R}})$	0.62294	1.32720	2.49557	3.98183	
	$\to\infty$		0.61983	1.33443	2.55590	4.26667	
5	10		1.30795	1.64243	2.10626	2.76323	3.71230
	100	$\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$	1.30034	1.64449	2.14401	2.90690	4.15187
	$\to\infty$		1.29955	1.64474	2.14823	2.92398	4.21053
	10		0.48929	0.91187	1.47111	2.06551	2.23756
	100	$\sigma_{\mathsf{sufl}}^{2}(D_{2n}^{\mathcal{R}})$	0.46442	0.92860	1.64749	2.70233	3.92519
	$\to\infty$		0.46189	0.93057	1.66828	2.78718	4.21053

Table 2: Exact and asymptotical values for $\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$ and $\sigma_{\mathsf{sufl}}^{2}(D_{2n}^{\mathcal{R}})$
for relations with $1 \le |T_{[}| \le |\mathcal{R}| \le 5$.

$\lvert\mathcal{R}\rvert$	n	$\lvert T_[\rvert$	1	$\lceil\frac{\lvert\mathcal{R}\rvert}{4}\rceil$	$\frac{\lvert\mathcal{R}\rvert}{2}$	$\lceil\frac{3\lvert\mathcal{R}\rvert}{4}\rceil$	$\lvert\mathcal{R}\rvert$
10	10	$\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$	1.14058	1.41839	1.79957	2.68132	3.60298
	100		1.13684	1.41946	1.82102	2.83122	4.04378
	$\to\infty$		1.13644	1.41957	1.82336	2.84900	4.10256
	10	$\sigma_{\mathsf{sufl}}^2(D_{2n}^{\mathcal{R}})$	0.21381	0.54447	1.01347	1.88943	2.12362
	100		0.20198	0.55127	1.10638	2.53566	3.81660
	$\to\infty$		0.20075	0.55203	1.11687	2.62173	4.10256
20	10	$\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$	1.06728	1.32106	1.77318	2.46325	3.55073
	100		1.06545	1.32250	1.79762	2.57895	3.99182
	$\to\infty$		1.06526	1.32266	1.80028	2.59241	4.05063
	10	$\sigma_{\mathsf{sufl}}^2(D_{2n}^{\mathcal{R}})$	0.09993	0.38904	0.93847	1.66690	2.07031
	100		0.09427	0.39533	1.03942	2.16155	3.76452
	$\to\infty$		0.09368	0.39601	1.05080	2.22487	4.05063
50	10	$\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$	1.02623	1.32422	1.75772	2.47605	3.52012
	100		1.02551	1.32747	1.78386	2.60006	3.96128
	$\to\infty$		1.02543	1.32782	1.78671	2.61453	4.02010
	10	$\sigma_{\mathsf{sufl}}^2(D_{2n}^{\mathcal{R}})$	0.03839	0.37172	0.89539	1.64838	2.03940
	100		0.03620	0.38376	1.00071	2.17339	3.73395
	$\to\infty$		0.03597	0.38505	1.01257	2.24113	4.02010
100	10	$\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$	1.01300	1.30572	1.75262	2.43480	3.51003
	100		1.01264	1.30904	1.77932	2.55271	3.95120
	$\to\infty$		1.01261	1.30940	1.78223	2.56642	4.01003
	10	$\sigma_{\mathsf{sufl}}^2(D_{2n}^{\mathcal{R}})$	0.01894	0.34303	0.88134	1.60367	2.02927
	100		0.01785	0.35496	0.98804	2.10131	3.72387
	$\to\infty$		0.01774	0.35623	1.00006	2.16505	4.01003

Table 3: Exact and asymptotical values for $\mu_{\mathsf{sufl}}(D_{2n}^{\mathcal{R}})$ and $\sigma_{\mathsf{sufl}}^2(D_{2n}^{\mathcal{R}})$ for relations with $\lvert\mathcal{R}\rvert \in \{10, 20, 50, 100\}$ and $\lvert T_[\rvert \in \left\{1, \lceil\frac{\lvert\mathcal{R}\rvert}{4}\rceil, \frac{\lvert\mathcal{R}\rvert}{2}, \lceil\frac{3\lvert\mathcal{R}\rvert}{4}\rceil, \lvert\mathcal{R}\rvert\right\}$.

5 Concluding Remarks

In this paper we have presented an algorithm that generates all words of a generalized Dyck language lexicographically. The Dyck language is defined by a relation $\mathcal{R}$ which describes the pairs of brackets that can be used. We introduced a function that computes from one Dyck word the next one according to the lexicographical order.

Further, we found a condition for the generalized Dyck language to be simply generated, which means that for every word it is possible to compute its successor by reading the suffix to be changed only. We saw that this condition depends on the relation $\mathcal{R}$ only and not on the length of the words. We introduced an algorithm that computes, whether the Dyck language – implied by the relation $\mathcal{R}$ – is simply generated or not. The running-time of that algorithm depends on the relation only, so it has a constant amount of time with respect to the length of the words.

Following a general approach to the lexicographical generation of all words of a formal language [Ke98], we computed the s-th moments, $s \geq 1$, of the random variable describing the length of the suffix of a word to be changed. In particular, we pointed out the mean value and the variance of the number of symbols to be changed in order to generate the successor of a Dyck word.

Acknowledgement

The author would like to thank Prof. Dr. Rainer Kemp, as it was his idea of how to compute the s-th moments of the random variable describing the length of the suffix to be changed.

References

[GKP94] R. L. Graham, D. E. Knuth and O. Patashnik, *Concrete Mathematics, 2nd ed.*, Addison-Wesley, 1994.

[GK82] D. H. Greene and D. E. Knuth, *Mathematics For The Analysis Of Algorithms, 2nd ed.*, Birkhäuser, 1982.

[Ha78] M. A. Harrison, *Introduction To Formal Language Theory*, Addison-Wesley, 1978.

[Ke84] R. Kemp, *Fundamentals Of The Average Case Analysis Of Particular Algorithms*, Wiley-Teubner, 1984.

[Ke96] R. Kemp, On The Average Minimal Prefix-Length Of The Generalized Semi-Dycklanguage, *RAIRO Theoretical Informatics and Applications* **30**, 1996, 6, 545-561.

[Ke98] R. Kemp, Generating Words Lexicographically: An Average-Case Analysis, *Acta Informatica* **35**, 1998, 17-89.

[Li96] J. Liebehenschel, Lexikographische Erzeugung der Dycksprachen mit mehreren Klammertypen, Diplomarbeit, Johann Wolfgang Goethe-Universität, Frankfurt am Main, Fachbereich Informatik, 1996.