平成28年度 言語・オートマトン 演習：生成文法と文脈自由言語 (解答・解説)

1. 講義中に与えた句構造文法 $G_1 = (N = \{S\}, \Sigma = \{a, b\}, P = \{S \to ab,\ S \to aSb)$ について次の各問に答えよ.

   (a) aaaabbbb $\in L(G_1)$ であることを示せ.

   (1 導出 $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbbb \Rightarrow aaaabbbb$ を構成すると aaaabbbb$\in L(G_1)$ であることを示される. CKY 法を学習した後であれば，演習 10 の解答と同様に示す.

   (b) aaabbaa$\notin L(G_1)$, aaaabb$\notin L(G_1)$ である理由を説明せよ.

   導出の列 $S \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow u_n = w \in \Sigma^*$ において u_i の右端に a が現れたとすると, P には $S \to ua$ $(u \in (N \cup \Sigma)^*$ が含まれなければならないが，そのような規則は P には含まれていないので $aaabbaa \notin L(G_1)$. 次に

   - $\sharp_a(w)$ で w に出現する a の個数を,
   - $\sharp_b(w)$ で w に出現する b の個数の和を,

   表すものとすると，各 i において $\sharp_a(u_i) = \sharp_b(u_i)$ であることは i に関する帰納法を用いれば容易に証明することができる. ここで $\sharp_a(aaaabb) = \sharp_b(aaaabb)$ なので $aaaabb \notin L(G_1)$.
   CKY 法を学習した後であれば，演習 10 の解答と同様に示す.

2. 講義中に与えた句構造文法 G_4 について次の各問に答えよ.

   (a) aaabbbccc$\in L(G_4)$ であることを示せ.

   文脈自由文法ではない（文脈依存文法) なので CKY 法は使えない. 講義中に説明した $aabbcc$ の導出と同様に導出を構成すればよい.

$$
\begin{aligned}
S &\Rightarrow aBCT \Rightarrow aBCABCT \Rightarrow aBCABCABC \Rightarrow aBACBCABC \Rightarrow aBABCCABC \\
&\Rightarrow aBABCACBC \Rightarrow aBABCABCC \Rightarrow aBABACBCC \Rightarrow aBABABCCC \\
&\cdots \Rightarrow aAABBBCCC \Rightarrow aaABBBCCC \Rightarrow \cdots \Rightarrow aaabbbccc
\end{aligned}
$$

   (b) aabbccc$\notin L(G_4)$, aaabbccc$\notin L(G_4)$ である理由を説明せよ.
   最初に aabbccc$\in L(G_4)$ である, つまり導出

$$
S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n = aabbccc
$$

   と仮定しよう. 生成規則で左辺が S であるようなもののなかか $R_1 : S \to aBC$ を最初に選択したとすると, w に至る導出は $S \Rightarrow aBC \Rightarrow abc$ だけである. したがって, 最初の導出は $R_2 : S \Rightarrow aBCT$ でなければならない. つぎに左辺が T である $R_3 : T \Rightarrow ABCT$ と $R_4 : T \Rightarrow ABC$ について考えると, 右辺に T が出現する規則は R_2 と R_3 だけであるから, R_4 が適用された後に R_3 が適用されることはなく, 一度 R_4 が適用されると, その後 $R_1, \ldots, R_4$ が適用されることはない. そこで $w_{k-1} \Rightarrow w_k$ において R_4 が適用されたとする. 各生成規則の左右両辺の長さに着目すると, $|w_i|$ は $k = 1, \ldots, k-1$ において 3 の倍数+1 であり, $i = k, \ldots n$ に

おいて 3 の倍数となることがわかる．一方 $|aabbccc|$ は 3 の倍数でないから, aabbccc$\notin L(G_4)$ と帰結できる．aaabbccc についても同じである．なお, この論法は abbbbcccc$\notin L(G_4)$ には使えない．

(別解) 非終端器号と終端記号からなる列 $w \in \{S, T, A, B, C\} \cup \{a, b, c\})^*$ に対して,

- $\sharp_{Aa}(w)$ で w に出現する A の個数と a の個数の和を,
- $\sharp_{Bb}(w)$ で w に出現する B の個数と b の個数の和を,
- $\sharp_{Cc}(w)$ で w に出現する C の個数と c の個数の和を

それぞれ表すものとする．　このとき

$S \Rightarrow^* w$ であれば $\sharp_{Aa}(w) = \sharp_{Bb}(w) = \sharp_{Cc}(w)$ である

ことを直接の導出の回数による帰納法で証明すれば, $\sharp_{Aa}(aabbccc) \neq \sharp_{Bb}(aabbccc)$ より aabbccc$\notin$ $L(G_4)$ と帰結できる.aaabbccc についても同じである．

3. 講義中に与えた句構造文法 G_2 について次の各問に答えよ．
   (a) $L(G_2) = \{$ ababbabbabb $\}$ であることを示せ．
   (b) ababbabbabb の構文木を描け．

   (略)

4. * 次の生成規則の集合 P を用いた句構造文法 $G = (\{S\}, \{\mathrm{a}, \mathrm{b}\}, P, S)$ について, (1)aaaaabbb$\in L(G)$ であることを示せ．(2) bbbaaaaa$\in L(G)$ であるか？ 理由をつけて答えよ．

$$\{S \to \mathrm{a}\ S,\ S \to S\ \mathrm{b},\ S \to \mathrm{a},\ S \to \mathrm{b}\}$$

(2) 直観的な説明: 導出 $S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n \in \Sigma^*$ において, w_i に対して生成規則 $S \to \mathrm{a}\ S$ を適用するということは a を S の左に挿入するということであり, 生成規則 $S \to\ S\mathrm{b}$ を適用するということは b を S の右に挿入するということである．

実は $L(G) = \{a^n b^m \mid n, m \in \mathbf{N}^+\}$ であり, $k \in \mathbf{N}^+$ について

$$\begin{aligned} L(G)|_k &= \{a^n b^m \mid n + m = k \text{ かつ } n, m \in \mathbf{N}^+\} \\ &= \{w \mid S \Rightarrow w_1 \Rightarrow w_2 \cdots \Rightarrow w_k\} \end{aligned}$$

である．このことは k に関する帰納法で示すことができる．実際, $k = 1$ のとき, w_1=a または w_1=b であるから, 上の等式は成り立つ．次に $k = i$ まで上の等式が成り立つと仮定して, $k = i + 1$ のときを考える．導出 $S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_k \Rightarrow w_{k+1}$ において, $w_1 = aS$ または $w_1 = Sb$ である．$w_1 = aS$ と仮定すると, $\underline{G \text{ は文脈自由文法だから}}$ $w_2 = aw'_2 \ldots w_{k+1} = aw'_{k+1}$ と表すことができる．このとき導出 $S \Rightarrow w'_2 \Rightarrow \cdots \Rightarrow w'_k \Rightarrow w'_{k+1}$ は長さが k だから, 帰納法の仮定より $w'_{k+1} \in L(G)|_k$ である．つまり $w_{k+1} \in \{aw \mid w \in L(G)|_k\}$ である．$w_1 = Sb$ の場合も同様であるから, $k = i + 1$ のときも上の等式は成り立つ．

なお $L(G)$ は正則言語であるが, G は右線形でも左線形でもないことに注意しよう．

5. 次の生成規則の集合 P を用いた句構造文法 $G = (\{S\}, \{\mathrm{i, e, a}\}, P, S)$ があいまいであることを示す文字列 $w \in L(G)$ を構成せよ．

$$\{S \to \mathrm{i}\ S,\ S \to \mathrm{i}\ S\ \mathrm{e}\ S,\ S \to \mathrm{a}\}$$

例えば $iaiae$ は 2 通りの構文木をもつ．なお，この文法は C 言語を祖とするのプログラミング言語における if 文と if-else 文を範としている．

6. 2 項演算子を引数の前に置いて表記する方法を**前置記法**あるいは**ポーランド記法** (Polish notation) という．たとえば，$a + b$, $c * (a + b)$ をポーランド記法で表すとそれぞれ $+\,a\,b$, $*\,c\ +a\,b$ となる．2 項演算子についてポーランド記法を用いると括弧を用いる必要がないことに注意しよう．

   (a) $((a + c) + b) * (a * (b + c))$ をポーランド記法で表せ．

   ポーランド記法では $*\ +\ +\,a\,c\,b\ *\ a\ +\,b\,c$ となる．

   (b) $\Sigma = \{a, b, c, +*\}$ とするとき，ポーランド記法で表された式を生成する文脈自由文法 G を与えよ．

   非終端記号は E だけを用いて $G = (\{E\}, \Sigma, P, E)$ とする．生成規則の集合 P は次の生成規則からなる:

$$E \to +E\,E, \quad E \to *E\,E, \quad E \to a, \quad E \to b, \quad E \to c$$

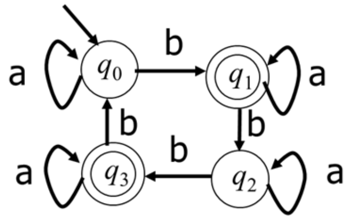
   (c) 前小問で与えた G はあいまいな文法であるか? (上述の注意はあいまいではない文法が構成できることを示唆している)

   曖昧ではない．直観的に説明すると，文字列 w を生成する導出 $S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n$ において，規則 $E \to +E\,E$ と $E \to *E\,E$ を適用する回数は w 中の $+$ と $*$ の個数で決まる．したがって，文法があいまいであるとしたら，2 つの規則の適用の順序を入れ換えることで 2 つの構文木ができているはずである．しかし実際に 2 つの規則の適用する回数を一定にして適用の順序を入れ換えると適用する非終端記号がどこであろうと w とは異なる文字列が生成される．したがって，G はあいまいではない．

   実は G は DPDA に変換できる．

   中置記法の文法では $E \Rightarrow E + E \Rightarrow E + E * E$ と $E \Rightarrow E * E \Rightarrow E + E * E$ のように，前者の導出における生成規則の適用順を入れ換え，しかも適用する非終端記号の位置を適当に変えれば同じ文字列を構成することがあり，このことが 2 つの導出が異なる構文木をつくる原因となっている．

7. 図に示す有限状態オートマトン M について，$L(M)$ を生成する正則文法を与えよ．



講義資料にある方法を直接用いれば

$$
\begin{aligned}
G &= (\{S, A, B, C\}, P, S) \\
P &= \left\{ \begin{array}{lll} S \to aS, & S \to bA, & S \to b, \\ A \to aA, & A \to bB, & \\ B \to aB, & B \to bC, & B \to b, \\ C \to aC, & C \to bS & \end{array} \right\}
\end{aligned}
$$

を得る．ここで状態 q_0, q_1, q_2, q_3 に対応する非終端記号をそれぞれ S, A, B, C とした．

8. ** 句構造文法 $G = (N, \Sigma, P, S)$ において，P の中のすべての生成規則が

$$A \to c, \qquad A \to Bc, \qquad A \to \varepsilon \qquad (A, B \in N,\ c \in \Sigma)$$

のいずれかの形をしているとき，**左正則文法**であるという[1]．ある言語 L が左正則文法で生成されることと正則言語であることは同値であることを証明せよ．
(演習問題 13 の解を一般的に説明する)

9. アルファベットを Σ とするとき，文字列 $w \in \Sigma^*$ を逆順に並べて得られる文字列を w^R と表すことにする．形式言語 $L_{Pal} = \{w \in \Sigma^* \mid w = w^R\}$ の要素を**回文 (palindrome)** という．以下では $\Sigma = \{a, b\}$ とする．

(a) $L(G_{Pal}) = L_{Pal}$ であるような文脈自由文法 G_{Pal} を与えよ．

「記号列の集合」の演習問題 3(b) で示した等式

$$L_{PAL} = \{u(u^R) \mid u \in \Sigma^*\} \cup \{ua(u^R) \mid u \in \Sigma^*\} \cup \{ub(u^R) \mid u \in \Sigma^*\}.$$

より，

$$G_{Pal} = (\{S\}, \Sigma, \{S \to aSa,\ S \to bSb,\ S \to a,\ S \to b,\ S \to \varepsilon\}, S)$$

であることは推測できるであろう．導出の段数に関する帰納法を用いて，$L(G_{Pal}) \subseteq L_{Pal}$ と $L(G_{Pal}) \supseteq L_{Pal}$ をそれぞれ証明することになる．（以下略)

(b) 前問で与えた G_{Pal} を (Chomsky 標準型でない場合) は Chomsky 標準型に変換せよ．

まず記号 a, b に対して，終端記号 A, B を導入して，$G'_{Pal} = (\{S, A, B\}, \Sigma, P', S)$, ここで

$$P = \{S \to ASA,\ S \to BSB,\ S \to A,\ S \to B,\ S \to \varepsilon\}).$$

[1]講義中で定義した正則文法を**右正則文法**ともよばれる．

さらに右辺 ASA, BSB をそれぞれ非終端記号 2 つにするために, 非終端記号 T, U を導入して $G''_{Pal} = (\{S, A, B, T, U\}, \Sigma, P'', S)$, ここで

$$P'' = \left\{ \begin{array}{l} S \to AT \\ T \to SA \\ S \to BU \\ U \to SB \\ S \to A \\ S \to B \\ S \to \varepsilon \\ A \to a \\ B \to b \end{array} \right\}.$$

さらに $S \to A$, $S \to U$ は非終端記号を書き換えているだけなので (単位規則, 問題 4 参照) なので, これらはそれぞれ $A \to a$, $U \to SB$ を用いて $S \to a$, $S \to SB$ に置き換えると, 求める Chomsky 標準形は $G'''_{Pal} = (\{S, A, B, T, U\}, \Sigma, P''', S)$, ここで

$$P''' = \left\{ \begin{array}{l} S \to AT \\ S \to BU \\ S \to a \\ S \to b \\ S \to \varepsilon \\ T \to SA \\ U \to SB \\ A \to a \\ B \to b \end{array} \right\}.$$

(c) CYK 法により aababaa $\in L_{Pal}$ であることを示せ. また, aababba $\notin L_{Pal}$ であることを示せ.

文字列 $aababaa$ については動的計画法の表は以下の通り構成され, その $(7, 1)$ 成分に S が含まれているので $aababaa \in L_{Pal}$ である.

	a	a	b	a	b	a	a
	1	2	3	4	5	6	7
1	S, A	S, A	S, B	S, A	S, B	S, A	S, A
2	T	U	T	U	T	T	
3	$\emptyset$	S	S	S	$\emptyset$		
4	$\emptyset$	U	T	S			
5	$\emptyset$	S	$\emptyset$				
6	$\emptyset$	T					
7	S						

一方, 文字列 $aababba$ については動的計画法を実行する表は以下の通りとなる.

	a	a	b	a	b	b	a
	1	2	3	4	5	6	7
1	S, A	S, A	S, B	S, A	S, B	S, B	S, A
2	T	U	T	U	U	T	
3	$\emptyset$	S	S	$\emptyset$	$\emptyset$		
4	$\emptyset$	U	U	$\emptyset$			
5	$\emptyset$	$\emptyset$	$\emptyset$				
6	$\emptyset$	$\emptyset$					
7	$\emptyset$						

CYK 法は導出の全ての可能性を網羅的に探索しているので$(7,1)$ 成分が空集合であることから $aababba \notin L_{Pal}$ と結論できる.

(d) G_{Pal} を Greibach 標準型に変換せよ.

非終端記号の順序を $A < B < S < T < U$ と決めておく. (この順序の決め方で結果は変わることがある). まず上の Chomsky 標準形の生成規則を $X \to Y\alpha$ $(X < Y)$ を満たすように A から順に変形する (上り).

$$\left\{ \begin{array}{cccc} & S \to aT & T \to aTA & U \to aTB \\ & S \to bU & T \to bUA & U \to bUB \\ A \to a & S \to a & T \to aA & U \to aB \\ B \to b & S \to b & T \to bA & U \to bB \\ & S \to \varepsilon & T \to A & U \to B \end{array} \right\}.$$

次に $X \to Y\alpha$ の Y を除去するのだが, この場合は適用の必要がなく, Greibach 標準形になっている.

なお非終端記号の順序を $U < T < S < B < A$ とすると, 上りにおいては何の走査をする必要もなく, 今度は下りで $X \to Y\alpha$ の Y が除去され, 上の Greibach 標準形を得る.

10. 正しい数式から括弧 (‘(’, ‘)’, ‘[’, ‘]’ など) だけを取り出して得られる言語を **Dyck 言語**とよぶ. 括弧では分かり辛いので, $\Sigma_{Dyck} = \{a, \overline{a}, b, \overline{b}\}$ としておく [2].Dyck 言語は文脈自由文法 $G_{Dyck} = (\{S\}, \Sigma_{Dyck}, P_{Dyck}, S\}$ で定義される. ここで,

$$P_{Dyck} \quad = \quad \{S \to SS,\ S \to aS\overline{a},\ S \to bS\overline{b},\ S \to \varepsilon\}$$

である.

(a) G_{Dyck} を Chomsky 標準型に変換せよ.

[2] マークアップ言語である HTML や XML における**タグ**に相当する.

講義中に説明した方法で Chomsky 標準形を生成すると以下のようになる．

$$\left\{\begin{array}{l} S \to SS \\ S \to AT \\ S \to BU \\ T \to SC \\ U \to SD \\ S \to \varepsilon \\ A \to a \\ B \to b \\ C \to \overline{a} \\ D \to \overline{b} \end{array}\right\}.$$

(b) CYK 法により $ab\overline{b}\overline{a}b\overline{b} \in L_{Dyck}$ であることを示せ．また，$ba\overline{b}\overline{a}b\overline{b} \notin L_{Dyck}$ であることを示せ．

上の Chomsky 標準型では直接には CKY 法を用いることができず, 工夫が必要である．CKY 法は与えられた文字列 w を 2 分することを繰り返して構文解析を行う．しかし空列の生成規則 $S \to \varepsilon$ が適用されたとすると，それは “2 分” することではわからない．そこで $L(G'_{Dyck}) = L(G_{Dyck} - \{\varepsilon\}$ であるような Chomsky 標準型の文法 G' を求める．規則の集合 P_{Dyck} の中で空列の生成規則 $S \to \varepsilon$ だけだから非終端記号 S に着目する．いま， P_{Dyck} による導出の列 $S = u_0 \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow u_n = w \in \Sigma^*$ において $u_i \quad (i \geq 1)$ に S が現れて, その S に $S \to \varepsilon$ が適用されて u_{i+1} が生成されたとしよう．その S は, $u_1 S \to SS$, $S \to aS\overline{a}$, $S \to bS\overline{b}$ のいすれかを $u_0, u_1, \ldots, u_{i-1}$ のいずれか u_j に適用した結果 u_i にあらわれているはずである．そして u_{i+1} はその S が単に消えているだけだから, 新たに生成規則 $S \to S$, $S \to a\overline{a}$,$S \to b\overline{b}$ を用意しておいて, u_j にそのどれかを適用しても u_{i+1} は得られているはずである．
(細かなことを言うと，文脈 “自由” であること＝構文解析木を描けること，が効いている)
ここで $S \to S$ は生成規則として冗長なのだからこれも除くとすると，G'_{Dyck} の生成規則は

$$P_{Dyck} \quad = \quad \{S \to SS,\ S \to aS\overline{a},\ S \to bS\overline{b},\ S \to a\overline{a},\ S \to bS\overline{b}\}$$

としてよい．$L(G'_{Dyck}) \not\ni \varepsilon$ であることは，問題 1(b) と同じ要領で示すことができる．G'_{Dyck} の Chomsky 標準型は

$$\left\{\begin{array}{l} S \to SS \\ S \to AT \\ S \to BU \\ T \to SC \\ T \to AC \\ U \to SD \\ U \to BD \\ A \to a \\ B \to b \\ C \to \overline{a} \\ D \to \overline{b} \end{array}\right\}.$$

となる．

同じことを (a) の結果の示した結果に対して行ってもよい．その場合は

$$\left\{\begin{array}{l} S \to SS \\ S \to AT \\ S \to BU \\ T \to \overline{a} \\ U \to \overline{b} \\ A \to a \\ B \to b \\ C \to \overline{a} \\ D \to \overline{b} \end{array}\right\}.$$

を得る．この文法を用いれば, 以下の動的計画法の表を得る.

	a	b	$\overline{b}$	$\overline{a}$	b	$\overline{b}$
	1	2	3	4	5	6
1	A	B	U, D	T, C	B	U, D
2	$\emptyset$	S	$\emptyset$	$\emptyset$	S	
3	$\emptyset$	T	$\emptyset$	$\emptyset$		
4	S	$\emptyset$	$\emptyset$			
5	$\emptyset$	$\emptyset$				
6	S					

	b	a	$\overline{b}$	$\overline{a}$	b	$\overline{b}$
	1	2	3	4	5	6
1	B	A	U, D	T, C	B	U, D
2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	S	
3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$		
4	$\emptyset$	$\emptyset$	$\emptyset$			
5	$\emptyset$	$\emptyset$				
6	$\emptyset$					

(c) G_{Dyck} を Greibach 標準型に変換せよ.

非終端記号の順序を $A < B < S < T < U$ と決めておく．(この順序の決め方で結果は変わることがある)．上りの変換は, まず S に関する規則の左再帰を除去してから, 左辺の左端の A, B を除去する.

$$\begin{array}{lcl} S \to AT & & S \to aT \\ S \to BU & & S \to bU \\ S \to ATZ & & S \to aTZ \\ S \to BUZ & \Rightarrow & S \to bUZ \\ Z \to S & & Z \to S \\ Z \to SZ & & Z \to SZ \\ S \to \varepsilon & & S \to \varepsilon \end{array}$$

T, U に関する規則については左辺の左端の S, C, D を除去する,

$$\begin{array}{ll} T \to aTC & U \to aTD \\ T \to bUC & U \to bUD \\ T \to aTZC & U \to aTZD \\ T \to bUZC & U \to bUZD \\ T \to \overline{a} & U \to \overline{b} \end{array}$$

下りの変換では Z に関する規則の左辺の S を消去することになる.

$$\begin{array}{l} Z \to aT \\ Z \to bU \\ Z \to aTZ \\ Z \to bUZ \\ Z \to aTZZ \\ Z \to bUZZ \\ Z \to \varepsilon \end{array}$$

(d) 通常の数式では括弧に階層が定義されている. 例えば, ('(', ')' は '[', ']' よりも階層が低いとされ, [()()] は許されるが, ([]) は許されない. そこで Σ_{Dyck} において, a と $\overline{a}$ の階層が b と $\overline{b}$ より低いと仮定して, G_{Dyck} を修正し, その Chomsky 標準型と Greibach 標準型を与えよ.

(略)

11. 文脈自由文法 $G = (N, \Sigma, P, S)$ において, $L(G)$ の要素の生成に用いられない非終端記号 X は**無効記号 (useless symbol)** とよばれる. 数学的には $S \Rightarrow \cdots \Rightarrow \alpha X \beta \Rightarrow \cdots \Rightarrow w \in \Sigma^*$ であるような導出が**構成できない**ような非終端記号 X が無効記号である.

文脈自由文法 $G_5 = (\{S, A, B, C\}, \{a, b\}, P_6, S)$ において無効記号を求め, $L(G_6) = L(G_6')$ なる G_1' で無効記号のないものを求めよ. ここで

$$P_6 = \left\{ \begin{array}{llll} S \to AB, & S \to CA, & A \to a, & \\ B \to BC, & B \to AB, & C \to aB, & C \to b \end{array} \right\}.$$

非終端記号 B に対して, B を書き換える 2 つの規則は両方とも B 自身が右辺に現れるので $S \Rightarrow \cdots \Rightarrow \alpha B \beta \Rightarrow \cdots \Rightarrow w \in \Sigma^*$ であるような導出は構成できないので無効記号である. P_1 から右辺に B が出現するような生成規則は取り除いても生成する言語は変わらないので, G_1' の規則は $P_1' = \{S \to CA, A \to a, C \to b\}$ とすればよい.

12. 生成規則が $X_1 \to X_2$ $(X_1, X_2 \in N)$ という形をしているとき, **単位生成規則 (unit production)** とよぶ. 文脈自由文法 $G_2 = (\{S, A, B\}, \{a, b\}, P_7, S)$ において $L(G_7) = L(G_7')$ なるで単位生成規則がないものを求めよ. もし無効記号がもし存在すればそれも除去すること. さらに得られた文法を Greibach 標準型に変換せよ. ここで

$$P_7 = \left\{ \begin{array}{llll} S \to ASB, & S \to \varepsilon, & A \to aAS, & A \to a, \\ B \to SbS, & B \to A, & B \to bb & \end{array} \right\}.$$

まず $S \Longrightarrow ASB \Rightarrow aSB \Rightarrow aB \Rightarrow abb$ という導出が構成できるので $S,\ A,\ B$ どれも無効記号ではない. 単位規則 $B \to A$ を除去すると

$$P_2' \;=\; \left\{ \begin{array}{llllllll} S \to ASB & , & S \to \varepsilon & , & A \to aAS & , & A \to a & , \\ B \to SbS & , & B \to aAS & , & B \to a & , & B \to bb & \end{array} \right\}.$$

順序 $A < B < S$ を仮定して Greibach 標準形に変形すると

$$P_2'' \;=\; \left\{ \begin{array}{lllllllllll} S \to aASSB & , & S \to aSB & , & S \to \varepsilon & , & A \to aAS & , & A \to a & , & \\ B \to aASSBbS & , & B \to aSBbS & , & B \to bS & , & B \to aAS & , & B \to a & , & B \to bb \end{array} \right\}.$$

13. 左正則文法 $G = (\{S, A\}, \{a, b\}, \{S \to Ab,\ A \to Sa, A \to a\}, S)$ を右正則文法に変換せよ.

Greibach 標準形を構成する時と同じ要領で変換する. 順序 $A < S$ を仮定すると上りの変換では

$$S \to ab,\ S \to Sab, A \to Sa, A \to a$$

と変換した後に左再帰を除去して

$$S \to ab,\ S \to abZ, Z \to ab, Z \to abZ, S \to ab, A \to Sa, A \to a$$

下りの変換を行えば

$$\begin{array}{l} S \to ab,\ S \to abZ, Z \to ab, Z \to abZ, S \to ab, \\ A \to aba,\ S \to abZa,\ S \to aba, A \to a \end{array}$$

となる. あとは Chomsky 標準形を構成する要領で右正則にすればよい.

なお, オートマトンを構成すればもっと簡単に変形できる.

14. これまでの演習問題で用いた文法とその結果を利用して以下の言語を受理する NPDA を構成せよ.

(a) L_{Pal}

(b) $L_{Dyck} = L(G_{Dyck})$

(c) $L(G_6)$

(d) $L(G_7)$

15. アルファベット Σ が与えられたとき, $\sharp \notin \Sigma$ なる新しい文字 $\sharp$ を用意して, $\{\Sigma \cup \{\sharp\}$ 上の言語 $L^{\sharp}_{Pal}$ を $L^{\sharp}_{Pal} = \{w\sharp(w^R) \,|\, w \in \Sigma^*\}$ と定義する. 以下では $\Sigma = \{a, b\}$ とする.

(a) 語 $w \in (\Sigma \cup \{\sharp\})^*$ で $w \in L^{\sharp}_{Pal}$ かつ $|w| = 5$ であるようなものを列挙せよ.

(b) $L^{\sharp}_{Pal}$ を空スタックで受理する NPDA を与えよ.

(c) 前問の L_{Pal} を受理するような DPDA は (空スタックによる受理であっても終了状態による受理であっても) 構成することはできない. その理由を述べよ.