7. 文脈自由言語の性質

実用上、有用

1. <u>文脈自由言語の標準形</u>
2. 文脈自由言語の反復補題

   例) $L=\{ 0^n1^n2^n \mid n \geqq 0 \}$ はCFLではない

3. 文脈自由言語の閉包性

   例) $L_1=\{ 0^n1^n2^m \mid n,m \geqq 0\}$ も $L_2=\{0^m1^n2^n \mid n,m \geqq 0\}$ もCFLだが、
   $L_1 \cap L_2$はCFLではない。

4. 文脈自由言語の決定問題
   - 所属性問題は効率よく解ける(が、単純ではない)
   - 決定不能な問題がある
     - 与えられた CFG は曖昧か？
     - 与えられた CFL は本質的に曖昧か？
     - 二つの CFL が等しいか？

- 構文木が単純
- プログラムによる処理が楽
- 形式的証明の場合わけが減る

7. 1. 文脈自由言語の標準形

A,B,C: 非終端記号
a: 終端記号
α : 0個以上の非終端記号列

- 二つの標準形
  1. チョムスキー(Chomsky)標準形
     - 次の二つの生成規則しか含まない: $\begin{cases} A \to BC \\ A \to a \end{cases}$

     ⇒ 導出木の形が単純
  2. グライバッハ(Greibach)標準形
     - 次の生成規則しか含まない: $A \to a\,\alpha$

     ⇒ 導出の回数＝語の長さ

ここでは
Chomsky標準形
のみ取り上げる

[定理] 任意の CFL L に対して、Lー{ ε }を生成する
それぞれの標準形の CFG が存在する。
（実際に構成することができる。）

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

- CFG に関して、有用な手法
  1. 無用な記号(useless symbol)の除去:
     開始記号$\overset{*}{\Rightarrow}$終端記号列に無関係な非終端記号や終端記号を除去する
  2. ε -規則(ε -production)の除去:
     $A \to \varepsilon$ の形の規則を除去する
  3. 単位規則(unit production)の除去:
     $A \to B$の形の規則を除去する

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

和文テキスト(p.284)では$S \underline{\Rightarrow} \alpha X \beta$となっているがこれは間違い

7.1.1. 無用な記号(useless symbol)の除去

[定義] 文法 $G=(V,T,P,S)$において、

- 記号 X ($\in V \cup T$) が有用(useful) である $\overset{\text{def}}{\Leftrightarrow}$ 導出 $S \overset{*}{\Rightarrow} \alpha X \beta \overset{*}{\Rightarrow} w$ ($\in T^*$) が存在する
  - $S=\alpha X \beta$ (つまり$S=X$) や $\alpha X \beta = w$ も含まれる点に注意
- 記号 X ($\in V \cup T$)が無用(useless)である $\overset{\text{def}}{\Leftrightarrow}$ Xが有用ではない

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.1. 無用な記号(useless symbol)の除去

[定義] 文法 $G=(V,T,P,S)$において、

- 記号 $X\ (\in V \cup T)$ が生成的(generating) である $\overset{\text{def}}{\Leftrightarrow}$ 導出 $X \overset{*}{\Rightarrow} w\ (\in T^*)$ が存在する
  - ✓ $X=w$ も含まれる点に注意

- 記号 $X\ (\in V \cup T)$が到達可能(reachable)である $\overset{\text{def}}{\Leftrightarrow}$ 導出 $S \overset{*}{\Rightarrow} \alpha X \beta$ が存在する
  - ✓ $S=\alpha X \beta$ も含まれる点に注意

★有用な記号=生成的かつ到達可能

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.1. 無用な記号(useless symbol)の除去

[アルゴリズムの概要] 文法 $G=(V,T,P,S)$において、

① 生成的でない記号を除去する

② 到着可能でない記号を除去する

★①と②は順序を入れ替えるとうまくいかない

[例] $S \to AB \mid a,\ A \to b$

生成的: S, A, a, b
到着可能: S, A, B, a, b

<u>①→②の順:</u>

①の結果
‘$S \to AB$’ を除去し、
$S \to a,\ A \to b$ となる。

②の結果、
$S \to a$ を得る。

<u>②→①の順:</u>

②の結果は不変

①の結果
‘$S \to AB$’ を除去し、
$S \to a,$ $\boxed{A \to b}$ となる。

A,bは無用

7.1.1. 無用な記号(useless symbol)の除去

[定理] CFG $G=(V,T,P,S)$ において、$L(G)\neq\Phi$とする。

1. Gが生成的でない記号を含むなら、その記号と、その記号を含む規則を削除する。結果として得られる文法を$G_2=(V_2,T_2,P_2,S)$とする。

2. G_2で到着可能でない記号を除去する。結果として得られる文法を$G_1=(V_1,T_1,P_1,S)$とする。

このときG_1は無用な記号を含まず、$L(G)=L(G_1)$である。

★$L(G)\neq\Phi$より、G_2, G_1 の開始記号は S のままである。

★生成的でない記号や到着可能でない記号をどうやって見つけるか、という点は後述。

7.1.1. 無用な記号(useless symbol)の除去

[定理] CFG $G=(V,T,P,S)$ において、$L(G)\neq\Phi$とする。

1. Gから生成的でない記号と、その記号を含む規則を削除する。結果として得られる文法を $G_2=(V_2,T_2,P_2,S)$とする。
2. G_2で到着可能でない記号を除去する。結果として得られる文法を$G_1=(V_1,T_1,P_1,S)$とする。

このときG_1は無用な記号を含まず、$L(G)=L(G_1)$である。

[証明(概略)]

① $V_1\cup T_1$ の任意の記号Xは生成的で到着可能であることと、

② $L(G)=L(G_1)$となることを示せばよい。

① X は 1 で除去されなかったので、G で生成的。
したがって G_2 でも生成的。よって G_1 でも生成的。
また 2 で除去されなかったので、G_2 で到着可能。
したがって G_1 でも到着可能。よって X は G_1 で有用。

7.1.1. 無用な記号(useless symbol)の除去

[定理] CFG $G=(V,T,P,S)$ において、$L(G)\neq\Phi$とする。

1. Gから生成的でない記号と、その記号を含む規則を削除する。結果として得られる文法を $G_2=(V_2,T_2,P_2,S)$とする。
2. G_2で到着可能でない記号を除去する。結果として得られる文法を$G_1=(V_1,T_1,P_1,S)$とする。

このときG_1は無用な記号を含まず、$L(G)=L(G_1)$である。

[証明(概略)]

① $V_1\cup T_1$ の任意の記号Xは生成的で到着可能であることと、

② $L(G)=L(G_1)$となることを $\underline{L(G)\subseteq L(G_1)}$と$\underline{L(G_1)\subseteq L(G)}$ で示す。

② $L(G_1)\subseteq L(G)$: 規則を削除しているだけなので、$L(G_1)\subseteq L(G)$は自明。

$L(G)\subseteq L(G_1)$: 任意の $w\in L(G)$ が $w\in L(G_1)$ を満たすことを示す。

仮定より、$S\overset{*}{\underset{G}{\Rightarrow}}w$ となる。この導出途中で現れる記号はすべて生成的でかつ到達可能であるので、この導出は G_1 の導出としても有効である。したがって $S\overset{*}{\underset{G_1}{\Rightarrow}}w$ であり、$w\in L(G_1)$。

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.2. [生成的な記号]と[到着可能な記号]の計算方法

1. 生成的記号の計算
   1. $G=(V,T,P,S)$に対して、
      1. 基礎: Tの各要素は生成的(自分を生成するので)。よって生成的記号の集合 $\mathcal{GS}$ を T で初期化; $\mathcal{GS} := T$
      2. 帰納: 規則 $A \rightarrow \alpha$ かつ α 中の記号がすべて生成的なら、$\mathcal{GS} := \mathcal{GS} \cup \{A\}$

         これは、$\alpha = \varepsilon$ の場合も含む点を注意する。
      3. 2が適用できる限り適用する。

[定理] 上記のアルゴリズムは正しく生成的な記号を計算する。

[略証] 生成的な記号だけが $\mathcal{GS}$ に加えられること
どの生成的な記号も $\mathcal{GS}$ に加えられること

それぞれ帰納法で

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.2. [生成的な記号]と[到着可能な記号]の計算方法

2. 到着可能な記号の計算
   1. $G=(V,T,P,S)$に対して、
      1. 基礎: S は自分から自分へ到着可能。よって $\mathcal{RS} := \{S\}$ と初期化
      2. 帰納: $A \in V \cup T$でAが到着可能なら、規則 $A \to \alpha$ であるすべてのα中の記号は到着可能。つまり
         $\mathcal{RS} := \mathcal{RS} \cup \{a\}$ for all a in α.
         これは、$\alpha = \varepsilon$ の場合も含む点を注意する。
      3. 2が適用できる限り適用する。

[定理] 上記のアルゴリズムは正しく到着可能な記号を計算する。

[略証] 到着可能な記号だけが $\mathcal{RS}$ に加えられること
どの到着可能な記号も $\mathcal{RS}$ に加えられること

帰納法で

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.2. [生成的な記号]と[到着可能な記号]の計算方法

例) $G=(V=\{S,A,B\},T=\{a,b\},P,S)$ で

P: $S \to AB \mid a,\ A \to b$

- 生成的記号の計算:
  1. a,bは生成的。
  2. $A \to b$ より A は生成的。$S \to a$ より S は生成的。$\therefore \mathcal{GS}=\{S,A,a,b\}$

$G_2=(\{S,A\},\{a,b\},P',S\}$ で P': $S \to a,\ A \to b$ となる。

- 到達可能な記号の計算:
  - Sは到達可能。
  - $S \to a$よりaは到達可能。$\therefore \mathcal{RS}=\{S,a\}$

$G_1=(\{S\},\{a\},P'',S)$で P'': $S \to a$となる。

無用な記号を含まない文法が得られた。

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.3. ε-規則の除去

目標:「言語LがCFLなら、$L-\{\varepsilon\}$を生成するε-規則を持たないCFGが存在する」ことを示す。

- ε-規則は便利だが、本質的ではない。(アルゴリズム的観点からは扱いがけっこう面倒)

- εを含む言語をどうしても表現したいのであれば、標準化した CFG $G=(V,T,P,S)$に $S\rightarrow\varepsilon$を例外的に追加すればよい。

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.3. ε -規則の除去

[定義] 変数A が消去可能(nullable) $\overset{\text{def}}{\Leftrightarrow}$ $A \overset{*}{\Rightarrow} \varepsilon$

消去可能な変数を求めるアルゴリズム:
[基礎] $A \rightarrow \varepsilon$ がGの規則ならば、Aは消去可能。
[帰納] $B \rightarrow C_1 C_2 \ldots C_k$が$G$の規則で、すべての$C_i$が消去可能ならば$B$は消去可能。

[定理] 上記のアルゴリズムは正しく消去可能な記号を計算する。

[略証]
消去可能な記号だけが見つけられること(見つかる順番に関する帰納法)
どの消去可能な記号も見つかること（導出の長さに関する帰納法）

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.3. ε -規則の除去

[定義] 変数A が消去可能(nullable) $\overset{\text{def}}{\Leftrightarrow}$ $A \overset{*}{\Rightarrow} \varepsilon$

消去可能な変数を求めたあと、ε -規則を含まない文法を構成する:

★ 変数Aが消去可能でも、$A \overset{*}{\Rightarrow} w$ という規則を残す必要がある

[アイデア] 消去可能な変数Aに対して、例えば

$B \rightarrow CAD$, $A \rightarrow \varepsilon$, (Aに関する他の規則)

という規則は

$B \rightarrow CD$, $B \rightarrow CAD$, ($A \rightarrow \varepsilon$ は削除), (Aに関する他の規則)

に書き換える必要がある。

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.3. ε -規則の除去

[定義] 変数A が消去可能(nullable) $\overset{\text{def}}{\Leftrightarrow}$ $A \overset{*}{\Rightarrow} \varepsilon$

消去可能な変数を求めたあと、ε -規則を含まない文法を構成する方法:

1. $A \to X_1 X_2 \ldots X_k$ $(k \geqq 1)$ を P に属する規則とし、$m \leqq k$個のX_iが消去可能であったとする。このとき、2^m通りの可能なX_iの消去方法を考え、これらの規則をすべて追加する。
2. $A \to \varepsilon$ の形の規則はすべて削除する

例) $A \to WXYZ$ のうち、X,Zが消去可能なら、
$A \to WY$
$A \to WXY$
$A \to WYZ$
$A \to WXYZ$
の4通りの消去方法を適用した規則を追加する。

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.3. ε-規則の除去

消去可能な変数を求めたあと、ε-規則を含まない文法を構成する方法:

1. $A \to X_1 X_2 \dots X_k$ ($k \geqq 1$) を P に属する規則とし、$m \leqq k$個の X_iが消去可能であったとする。このとき、2^m通りの可能な X_iの消去方法を考え、これらの規則をすべて追加する。
2. $A \to \varepsilon$ の形の規則はすべて削除する

[定理] CFG Gから上記のアルゴリズムでε-規則を含まない CFG G_1 を構成すると、$L(G_1)=L(G)-\{\varepsilon\}$である。

[証明] 省略

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.3. ε-規則の除去

例) 規則が

$S \to AB$
$A \to aAB \mid \varepsilon$
$B \to bBB \mid \varepsilon$

の文法

テキスト改

ステップ1:
消去可能変数を見つける

$A \to \varepsilon$, $B \to \varepsilon$ … A,B
$S \to AB$ … S

∴S,A,Bは消去可能変数

ステップ2: 個々の規則の書き換え

- $S \to AB$
  $S \to A$, $S \to B$, $S \to AB$
- $A \to aAB$
  $A \to a$, $A \to aA$, $A \to aB$, $A \to aAB$
- $B \to bBB$
  $B \to b$, $B \to bB$, $B \to bBB$

ステップ3: 最終的な規則

$S \to A \mid B \mid AB$
$A \to a \mid aA \mid aB \mid aAB$
$B \to b \mid bB \mid bBB$

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.4. 単位規則の除去

$A \to B$の形の単位規則(unit production)を除去するとき、

- $A \to B, B \to C, C \to A$ といった、循環的なケースがある
- $A \to BC, C \to \varepsilon$ なら $A \overset{*}{\Rightarrow} B$ となりうる

ので、単に展開して削除するだけではうまくいかないことがある。

[準備] 「単位規則だけを使って $A \overset{*}{\Rightarrow} B$ となるペア(A,B)」
(単位ペア(unit pair)と呼ぶ)を以下の方法ですべて見つける:

[基礎] どの変数Aについても(A,A)は単位ペア

[帰納] (A,B)が単位ペアのとき、
$B \to C$が単位規則なら(A,C)も単位ペア

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.4. 単位規則の除去

[準備] 「単位規則だけを使って $A \overset{*}{\Rightarrow} B$ となるペア(A,B)」(単位ペア(unit pair)と呼ぶ)を以下の方法ですべて見つける:

[基礎] どの変数Aについても(A,A)は単位ペア

[帰納] (A,B)が単位ペアのとき、
$B \rightarrow C$が単位規則なら(A,C)も単位ペア

[定理] CFG Gで上記のアルゴリズムですべての単位ペアを見つけることができる。

[証明] 省略

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.4. 単位規則の除去

[単位規則の除去]

1. 単位ペアをすべて見つける
2. すべての
   - 単位ペア(A,B)
   - 単位規則ではない規則$B\to\alpha$

   に対して、規則$A\to\alpha$を追加する
3. 単位規則を削除する

[定理] CFG Gから上記のアルゴリズムで単位規則を含まない CFG G_1 を構成すると、$L(G_1)=L(G)$である。

[証明] 省略

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.4. 単位規則の除去

例) 規則が

$$I \to a \mid (E)$$
$$F \to F \times I \mid I$$
$$E \to E+F \mid F$$

の文法(スタート記号はE)

ステップ1:
単位ペアを見つける
$F \to I$, $E \to F$ … (F,I), (E,F)
さらに (E,I) も
∴(F,I), (E,F), (E,I)が単位ペア

ステップ2: 追加すべき規則
- (F,I)より
  $F \to a \mid (E)$
- (E,F)より
  $E \to F \times I \mid a \mid (E)$
- (E,I)より
  $E \to F \times I \mid a \mid (E)$

ステップ3: 最終的な規則
$I \to a \mid (E)$
$F \to F \times I \mid a \mid (E)$
$E \to E+F \mid F \times I \mid a \mid (E)$

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

7.1.1～7.1.4. まとめ

[単純化アルゴリズムまとめ]

1. ε-規則を削除する
2. 単位規則を削除する
3. 無用な記号を除く

という順番で変換を適用すると、以下の定理が得られる。

[定理] CFG G がε以外の語を少なくとも1つ生成するとする。このとき上記のアルゴリズムを適用して作成した CFG G_1 は、$L(G_1)=L(G)-\{\varepsilon\}$であり、ε-規則と単位規則は持たず、無用な記号も持たない。

[証明] 省略(適用順序は大切であることに注意)

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

A,B,C: 非終端記号
a: 終端記号
α: 0個以上の非終端記号列

7.1.5. Chomsky 標準形

1. チョムスキー(Chomsky)標準形
   - 次の二つの生成規則しか含まない:

$$\begin{cases} A \rightarrow BC \\ A \rightarrow a \end{cases}$$

7.1.1.～7.1.4.のまとめ: 単純化した CFG G は
$A \rightarrow \varepsilon$ と$A \rightarrow B$の形の規則は含まない

単純化した CFG $G=(V,T,P,S)$ の規則 P は
(1) $A \rightarrow a$ OK
(2) $A \rightarrow X_1 X_2 \ldots X_k$

$$\begin{cases} X_i \in V \cup T \\ k \geqq 2 \end{cases}$$

の形の規則のみを含む。
これをChomskyの標準形に変換する。

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

A,B,C: 非終端記号
a: 終端記号
α: 0個以上の非終端記号列

7.1.5. Chomsky 標準形

1. チョムスキー(Chomsky)標準形
   - 次の二つの生成規則しか含まない:
     $$\begin{cases} A \to BC \\ A \to a \end{cases}$$

(2) $A \to X_1 X_2 \ldots X_k \quad \begin{cases} X_i \in V \cup T \\ k \geqq 2 \end{cases}$

[手順1] X_iが終端記号aなら、新たな非終端記号X_i'を導入し、

$$A \to X_1 X_2 \ldots X_{i-1} X_i' X_{i+1} \ldots X_k$$

$$X_i' \to a$$

OK

とする。この処理をすべてのX_iに適用してラベルをつけかえると…

(2') $A \to X_1 X_2 \ldots X_k \quad \begin{cases} X_i \in V \\ k \geqq 2 \end{cases}$

となる。

k=2もOK

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

A,B,C: 非終端記号
a: 終端記号
α: 0個以上の非終端記号列

7.1.5. Chomsky 標準形

1. チョムスキー(Chomsky)標準形
   - 次の二つの生成規則しか含まない: $\begin{cases} A \to BC \\ A \to a \end{cases}$

(2') $A \to X_1 X_2 \ldots X_k \begin{cases} X_i \in V \\ k \geqq 3 \end{cases}$

[手順2] 新たな非終端記号$Y_1, Y_2, \ldots, Y_{k-2}$を導入し、

$$A \to X_1 Y_1$$
$$Y_1 \to X_2 Y_2,\ Y_2 \to X_3 Y_3,\ \ldots,\ Y_{k-3} \to X_{k-2} Y_{k-2}$$
$$Y_{k-2} \to X_{k-1} X_k$$

OK

とする。

これですべて Chomsky の標準形のどちらかのタイプに変換できた。

7. 1. 文脈自由言語の標準形 (Chomsky標準形)

A,B,C: 非終端記号
a: 終端記号
α : 0個以上の非終端記号列

7.1.5. Chomsky 標準形

1. チョムスキー(Chomsky)標準形
   - 次の二つの生成規則しか含まない:

$$\begin{cases} A \to BC \\ A \to a \end{cases}$$

[まとめ] 任意の CFL L に対して、$L-\{\varepsilon\}$を生成する Chomsky標準形の CFG が存在する。

$L(G)=L$を満たす任意のCFG Gから実際に<u>構成できる</u>。

[おまけ] 任意の CFL L に対して、$L-\{\varepsilon\}$を生成する Greibach標準形の CFG が存在する。
(同様に構成的に示すことができる。)

7. 1. 文脈自由言語の標準形: 演習問題(9)

[問題] CFG G=({P}, {0,1}, X, P) ただし

X: $P \rightarrow \varepsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1$

とする。この文法をChomsky標準形に直せ。