7. 文脈自由言語の性質

1. 文脈自由言語の標準形
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 $L_2=\{0^m1^n2^n \mid n,m \geqq 0\}$
   $L_1 \cap L_2$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？

実用上、有用

• 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少ない

7. Property of Context Free Language

1. Normal Forms for CFL
2. Pumping lemma for CFL
   Ex) $L=\{ 0^n1^n2^n \mid n \geqq 0 \}$ is *not* CFL.
3. Closure property of CFL
   Ex) While $L_1=\{ 0^n1^n2^m \mid n,m \geqq 0\}$ and $L_2=\{0^m1^n2^n \mid n,m \geqq 0\}$ are CFLs,
   $L_1 \cap L_2$ is not a CFL.
4. Decision problems for CFL
   – Membership problem can be solved efficiently (but not so simple…)
   – Undecidable problems
     • Is given CFG ambiguous?
     • Is given CFL inherently ambiguous?
     • Are two CFLs the same language?

Useful from practical viewpoint

• Simple parse tree
• Easy for programming
• Simple proof for theoretical results

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、$L-\{\varepsilon\}$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. Normal forms for CFL

A,B,C: Nonterminal
a: Terminal
α : 0 or more nonterminals

➢ Two major normal forms
1. Chomsky Normal Form
   • consists of the following two types: $\begin{cases} A \to BC \\ A \to a \end{cases}$
   ⇨ Simple parse tree
2. Greibach Normal Form
   • consists of the following type: $A \to a\alpha$
   ⇨ number of derivations ＝ length of the word

[Theorem] For any CFL L, there are the normal forms that generate $L-\{\varepsilon\}$.
(Constructively proved)

Here we show Chomsky Normal Form

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. Normal forms for CFL (Chomsky Normal Form)

➢ Useful techniques for general CFG
1. Remove useless symbols:
   Remove the symbols which have no relation to any derivation
   ‘Start symbol $\overset{*}{\Rightarrow}$ Terminals’
2. Remove ε-productions:
   Remove the production ‘$A \to \varepsilon$’
3. Remove unit productions:
   Remove the production ‘$A \to B$’

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$S \Rightarrow \alpha X \beta$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. Normal forms for CFL (Chomsky Normal Form)

Japanese text says that $S \Rightarrow \alpha X \beta$ in p.284 which should be typo.

7.1.1. Remove useless symbols

[Definition]

For a grammar $G=(V,T,P,S)$,

- Symbol X ($\in V \cup T$) is useful $\overset{\text{def}}{\Leftrightarrow}$ There exists a derivation $S \overset{*}{\Rightarrow} \alpha X \beta \overset{*}{\Rightarrow} w$ ($\in T^*$)
  - Note: S can be $\alpha X \beta$ (or $S=X$), and $\alpha X \beta$ can be w.
- Symbol X ($\in V \cup T$) is useless $\overset{\text{def}}{\Leftrightarrow}$ X is not useful.

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

★有用な記号=生成的かつ到達可能

7. 1. Normal forms for CFL (Chomsky Normal Form)

7.1.1. Remove useless symbols

[Definition] For a grammar $G=(V,T,P,S)$,

- Symbol X ($\in V \cup T$) is generating $\overset{\text{def}}{\Leftrightarrow}$ There is a derivation $X \overset{*}{\Rightarrow} w$ for some $w \in T^*$
  - Note: X can be w
- Symbol X ($\in V \cup T$) is reachable $\overset{\text{def}}{\Leftrightarrow}$ There is a derivation $S \overset{*}{\Rightarrow} \alpha X \beta$

★Useful symbol = generating & reachable symbol

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 到達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
到達可能: S, A, B, a, b

①→②の順:
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。

②→①の順:
②の結果は不変
①の結果
'$S \to AB$' を除去し、
$S \to a$, $A \to b$ となる。

A,bは無用

7. 1. Normal forms for CFL (Chomsky Normal Form)

7.1.1. Remove useless symbols

[Outline of Algorithm] For a CFL $G=(V,T,P,S)$,

① Remove non-generating symbols
② Remove non-reachable symbols

★We cannot exchange ① and ②.

[Ex] $S \to AB \mid a$, $A \to b$

Generating: S, A, a, b
Reachable: S, A, B, a, b

①→②:
①: remove '$S \to AB$', and we have $S \to a$, $A \to b$
②: just $S \to a$.

②→①:
② has no effect
①: remove '$S \to AB$', and we have $S \to a$, $A \to b$.

A and b are useless

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で到達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達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. Remove useless symbols

[Theorem] Let $G=(V,T,P,S)$ be a CFG with $L(G)\neq\Phi$.

1. Remove all non-generating symbols and the rules containing them if G contains. Let $G_2=(V_2,T_2,P_2,S)$ be the resultant grammar.
2. Remove all non-reachable symbols in G_2. Let $G_1=(V_1,T_1,P_1,S)$ be the resultant grammar.

Then, G_1 contains no useless symbols and $L(G)=L(G_1)$.

★Since $L(G)\neq\Phi$, the start symbols of G_2 and G_1 are still S.
★We will consider how can we find those symbols, later.

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で到達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は生成的で到達可能であることと、
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 で到達可能。
したがって G_1 でも到達可能。よって X は G_1 で有用。

7.1.1. Remove useless symbols

[Theorem] Let $G=(V,T,P,S)$ be a CFG with $L(G)\neq\Phi$.

1. Remove all non-generating symbols and the rules containing them if G contains. Let $G_2=(V_2,T_2,P_2,S)$ be the resultant grammar.
2. Remove all non-reachable symbols in G_2. Let $G_1=(V_1,T_1,P_1,S)$ be the resultant grammar.

Then, G_1 contains no useless symbols and $L(G)=L(G_1)$.

[Proof (Sketch)] We show
① Any symbol X in $V_1\cup T_1$ is generating & reachable, and
② $L(G)=L(G_1)$.

① Since X was not removed in step 1, that is generating in G, and hence so is in G_2 . Thus X is generating also in G_1.
Since X was not removed in step 2, that is reachable in G_2, and hence so is in G_1. Thus X is useful in G_1.

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で到達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は生成的で到達可能であることと、
② $L(G)=L(G_1)$となることを $\underline{L(G)\subseteq L(G_1)}$<u>と</u>$\underline{L(G_1)\subseteq L(G)}$ 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、$w\in L(G_1)$。

7.1.1. Remove useless symbols

[Theorem] Let $G=(V,T,P,S)$ be a CFG with $L(G)\neq\Phi$.

1. Remove all non-generating symbols and the rules containing them if G contains. Let $G_2=(V_2,T_2,P_2,S)$ be the resultant grammar.
2. Remove all non-reachable symbols in G_2. Let $G_1=(V_1,T_1,P_1,S)$ be the resultant grammar.

Then, G_1 contains no useless symbols and $L(G)=L(G_1)$.

[Proof (Outline)] We show
① Any symbol X in $V_1\cup T_1$ is generating & reachable, and
② $L(G)=L(G_1)$ <u>by proving</u> $\underline{L(G)\subseteq L(G_1)}$ <u>and</u> $\underline{L(G_1)\subseteq L(G)}$.

② $L(G_1)\subseteq L(G)$: Since rules are just removed, hence $L(G_1)\subseteq L(G)$ is clear.
$L(G)\subseteq L(G_1)$: We show any $w\in L(G)$ satisfies $w\in L(G_1)$.
By assumption, $S\overset{*}{\underset{G}{\Rightarrow}}w$. All symbols appearing on this derivation are generating and reachable. Hence the derivation is also available in G_1.
Hence $S\overset{*}{\underset{G_1}{\Rightarrow}}w$, and $w\in L(G_1)$。

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[到達可能な記号]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. Normal forms for CFL (Chomsky Normal Form)

7.1.2. Computation of [generating symbols] & [reachable symbols]

1. Generating symbols:
   1. For a CFG $G=(V,T,P,S)$,
      1. Base: Each element in T is generating (since it generates itself). Hence, the set $\mathcal{GS}$ of generating symbols is initialized by T; $\mathcal{GS} := T$
      2. Induction: If all symbols in α is generating in a rule $A\rightarrow\alpha$, we update $\mathcal{GS} := \mathcal{GS} \cup \{A\}$.
         Note: It is applied for $\alpha=\varepsilon$.
      3. Repeat step 2 while it can be applied.

[Theorem] The algorithm surely computes the set of generating symbols.

[Proof (Sketch)]
Only generating symbols are added to $\mathcal{GS}$.
Any generating symbol is added to $\mathcal{GS}$.

They can be proved by simple inductions

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[到達可能な記号]の計算方法

2. 到達可能な記号の計算
   1. $G=(V,T,P,S)$に対して、
      1. 基礎: S は自分から自分へ到達可能。よって $\mathcal{RS} := \{S\}$ と初期化
      2. 帰納: $A\in V\cup T$でAが到達可能なら、規則 $A\rightarrow\alpha$ であるすべての α 中の記号は到達可能。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達可能な記号を計算する。

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

帰納法で

7. 1. Normal forms for CFL (Chomsky Normal Form)

7.1.2. Computation of [generating symbols] & [reachable symbols]

2. Reachable symbols:
   1. For a CFG $G=(V,T,P,S)$,
      1. Base: Since S is reachable from S to S, initialize $\mathcal{RS} := \{S\}$.
      2. Induction: For reachable $A\in V\cup T$, all symbols in α for a rule $A\rightarrow\alpha$ is reachable. Namely,
         $\mathcal{RS} := \mathcal{RS} \cup \{a\}$ for all a in α.
         Note that α can be ε.
      3. Repeat 2 while it can be applied.

[Theorem] The algorithm surely computes the set of reachable symbols.

[Proof (Sketch)]
Only reachable symbols are added to $\mathcal{RS}$.
Any reachable symbol is added to $\mathcal{RS}$.

They can be proved by simple inductions

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[到達可能な記号]の計算方法

例) $G=(V=\{S,A,B\},T=\{a,b\},P,S)$ で

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

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

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

- 到達可能な記号の計算:
  – Sは到達
  – $S\rightarrow a$より

無用な記号を含まない文法が得られた。

7. 1. Normal forms for CFL (Chomsky Normal Form)

7.1.2. Computation of [generating symbols] & [reachable symbols]

Ex) For a CFG $G=(V=\{S,A,B\},T=\{a,b\},P,S)$ with

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

- Computation of generating symbols:
  1. a and b are generating.
  2. $A\rightarrow b$ implies that A is generating. $S\rightarrow a$ implies that S is generating.
     $\therefore \mathcal{GS}=\{S,A,a,b\}$

We have $G_2=(\{S,A\},\{a,b\},P',S\}$ with P': $S\rightarrow a$, $A\rightarrow b$.

- Computation of reachable symbols:
  – S is reachable.
  – $S\rightarrow a$ implies that a is reachable. $\therefore \mathcal{RS}=\{S,a\}$

We have $G_1=(\{S\},\{a\},P'',S)$ with P'': $S\rightarrow a$.

The grammar G_1 contains no useless symbols.

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 $S\to\varepsilon$ を例外的に追加すればよい。

7. 1. Normal forms for CFL (Chomsky Normal Form)

7.1.3. Remove ε-productions

Goal: We show 'for any CFL L, there is a CFG that generates the language $L-\{\varepsilon\}$, and it has no ε-productions.'

- ε-production is useful, but it is not essential for languages. (From the algorithmic viewpoint, handling ε-production is troublesome.)
- You can add the special rule '$S\to\varepsilon$' to the standardized CFG $G=(V,T,P,S)$ as an exception rule to add ε to the language.

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] $A\to\varepsilon$ がGの規則ならば、Aは消去可能。
[帰納] $B\to C_1C_2\ldots C_k$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. Normal forms for CFL (Chomsky Normal Form)

7.1.3. Remove ε-productions

[Definition] A nonterminal A is nullable $\overset{\text{def}}{\Leftrightarrow}$ $A\overset{*}{\Rightarrow}\varepsilon$

Algorithm for finding nullable rules:
[Base] A is nullable if $A\to\varepsilon$ is a rule of G.
[Induction] B is nullable if $B\to C_1C_2\ldots C_k$ is a rule of G, and all C_i are nullable.

[Theorem] The algorithm surely computes all nullable nonterminals.

[Proof (Sketch)]
Only nullable nonterminals are found (induction for the order of found)
Any nullable nonterminal is found (induction for the number of derivations)

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
$B\to CAD$, $A\to\varepsilon$, (Aに関する他の規則)
という規則は
$B\to CD$, $B\to CAD$, ($A\to\varepsilon$ 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. Normal forms for CFL (Chomsky Normal Form)

7.1.3. Remove ε-productions

[Definition] A nonterminal A is nullable $\overset{\text{def}}{\Leftrightarrow}$ $A\overset{*}{\Rightarrow}\varepsilon$

After finding all nullable nonterminals, we construct a CFG that contains no ε-productions:

★ We have to remain the rule $A\overset{*}{\Rightarrow}w$ even if A is nullable.

[Idea] For a nullable nonterminal A, for example, we have to replace
$B\to CAD$, $A\to\varepsilon$, (Other rules for A)
by
$B\to CD$, $B\to CAD$, (Remove $A\to\varepsilon$), (Other rules for A).

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) $\overset{def}{\Leftrightarrow}$ $A \overset{*}{\Rightarrow} \varepsilon$

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:

1. $A \to X_1X_2 \ldots X_k$ $(k \geqq 1)$ 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。

31/54

7. 1. Normal forms for CFL (Chomsky Normal Form)

7.1.3. Remove ε -productions

[Definition] A nonterminal A is nullable $\overset{def}{\Leftrightarrow}$ $A \overset{*}{\Rightarrow} \varepsilon$

After finding all nullable nonterminals, we construct a CFG that contains no ε -productions as follows:

1. Let $A \to X_1X_2 \ldots X_k$ $(k \geqq 1)$ be a rule in P such that $m \leqq k$ X_is are nullable. Then, we apply all possible 2^m ways to remove X_is, and add them as new rules.
2. Remove the rule $A \to \varepsilon$.

Ex) For $A \to WXYZ$, suppose it has two nullable X and Z. Then, we add four possible rules: $A \to WY$, $A \to WXY$, $A \to WYZ$, $A \to WXYZ$

32/54

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. $A \to X_1X_2 \ldots X_k$ $(k \geqq 1)$ 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

[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

33/54

7. 1. Normal forms for CFL (Chomsky Normal Form)

7.1.3. Remove ε -productions

After finding all nullable nonterminals, we construct a CFG that contains no ε -productions as follows:

1. Let $A \to X_1X_2 \ldots X_k$ $(k \geqq 1)$ be a rule in P such that $m \leqq k$ X_is are nullable. Then, we apply all possible 2^m ways to remove X_is, and add them as new rules.
2. Remove the rule $A \to \varepsilon$.

[Theorem] For any given CFG G, construct the CFG G_1 without ε -production by the algorithm. Then we have $L(G_1)=L(G)-\{\varepsilon\}$.

[Proof] Omitted.

34/54

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

35/54

7. 1. Normal forms for CFL (Chomsky Normal Form)

7.1.3. Remove ε -productions

Ex.) For the rules

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

modified sample in text

Step 1:
Find all nullable nonterminals:
$A \to \varepsilon$, $B \to \varepsilon$ … A,B
$S \to AB$ … S
∴S,A,B are nullable.

Step 2: Replace rules:
- $S \to AB$
  - $S \to A$, $S \to B$, $S \to AB$
- $A \to aAB$
  - $A \to a$, $A \to aA$, $A \to aB$, $A \to aAB$
- $B \to bBB$
  - $B \to b$, $B \to bB$, $B \to bBB$

Step 3: Final rules:
$S \to A \mid B \mid AB$
$A \to a \mid aA \mid aB \mid aAB$
$B \to b \mid bB \mid bBB$

36/54

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. Normal forms for CFL (Chomsky Normal Form)

7.1.4. Remove unit productions

When we remove unit production $A\to B$, we have:

- Cyclic case: $A\to B, B\to C, C\to A$
- It can be $A\overset{*}{\Rightarrow}B$ when $A\to BC$ and $C\to\varepsilon$

Hence, 'simple expansion and remove' do not work.

[Preparation] We first find all unit pairs, which are pairs (A,B) such that $A\overset{*}{\Rightarrow}B$ by only using unit productions:

[Base] (A,A) is a unit pair for any nonterminal.

[Induction] For a unit pair (A,B), if $B\to C$ is unit production,
(A,C) is also unit pair.

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、
$B\to C$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. Normal forms for CFL (Chomsky Normal Form)

7.1.4. Remove unit productions

[Preparation] We first find all unit pairs, which are pairs (A,B) such that $A\overset{*}{\Rightarrow}B$ by only using unit productions:

[Base] (A,A) is a unit pair for any nonterminal.

[Induction] For a unit pair (A,B), if $B\to C$ is unit production,
(A,C) is also unit pair.

[Theorem] For any CFG G, the algorithm finds all unit pairs.

[Proof] Omitted.

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. Normal forms for CFL (Chomsky Normal Form)

7.1.4. Remove unit productions

[Remove unit productions]

1. Find all unit pairs
2. For all possible
   - unit pair (A,B)
   - non-unit production $B\to\alpha$

   add the rule $A\to\alpha$.
3. Remove all unit productions.

[Theorem] Let G_1 be the CFG obtained from a CFG G by the algorithm. Then, we have $L(G_1)=L(G)$.

[Proof] Omitted

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
  $E \to F \times I$
- (E,I)より
  $E \to a \mid (E)$

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. Normal forms for CFL (Chomsky Normal Form)

7.1.4. Remove unit productions

Ex) For a grammar with rules

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

(start symbol; E)

Step 1:
Find all unit pairs
$F \to I$, $E \to F$ … (F,I), (E,F)
and (E,I) is.
∴(F,I), (E,F), (E,I) are unit pairs.

Step 2: rules should be added
- (F,I) implies
  $F \to a \mid (E)$
- (E,F) implies
  $E \to F \times I$
- (E,I) implies
  $E \to a \mid (E)$

Step 3: Final rules
$I \to a \mid (E)$
$F \to F \times I \mid a \mid (E)$
$E \to E{+}F \mid F \times I \mid a \mid (E)$

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、$\varepsilon$ -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. Normal forms for CFL (Chomsky Normal Form)

7.1.1～7.1.4. Summary

[Outline of Simplify Algorithm]
Perform the following operations in the following ordering,
1. Remove ε -productions
2. Remove unit productions
3. Remove useless symbols

we have the following theorem.

[Theorem] Let G be a CFG that generates at least one word except ε. Then, the CFG G1 obtained from G by the algorithm satisfies (1) $L(G_1)=L(G)-\{\varepsilon\}$, (2) no ε -productions, (3) no unit productions, and (4) no useless symbols.

[Proof] Omitted (Note that the ordering is crucial.)

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

7.1.1.～7.1.4.のまとめ: 単純化した CFG G は $A \to \varepsilon$ と$A \to B$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) $A \to a$ OK
(2) $A \to X_1X_2 \ldots X_k$ $\begin{cases} X_i \in V \cup T \\ k \geqq 2 \end{cases}$
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. Normal forms for CFL (Chomsky Normal Form)

A,B,C: Nonterminals
a: terminal
α : 0 or more nonterminals

7.1.5. Chomsky Normal Form

1. Chomsky Normal Form
   - consists of the following two types: $\begin{cases} A \to BC \\ A \to a \end{cases}$

7.1.1.～7.1.4. Summary: Simplified CFG G has no rules with form $A \to \varepsilon$ and $A \to B$

Simplified CFG $G=(V,T,P,S)$ has the set of rules P, which has just two types of rules:
(1) $A \to a$ OK
(2) $A \to X_1X_2 \ldots X_k$ $\begin{cases} X_i \in V \cup T \\ k \geqq 2 \end{cases}$
We modify them to Chomsky Normal Form.

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) $A \to X_1X_2\ldots X_k$ $\begin{cases} X_i \in V \cup T \\ k \geqq 2 \end{cases}$

[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、

$A \to X_1X_2\ldots X_{i-1}X_i'X_{i+1}\ldots X_k$
$X_i' \to a$ 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') $A \to X_1X_2\ldots X_k$ $\begin{cases} X_i \in V \\ k \geqq 2 \end{cases}$ となる。 k=2もOK

7. 1. Normal forms for CFL (Chomsky Normal Form)

A,B,C: Nonterminals
a: terminal
α : 0 or more nonterminals

7.1.5. Chomsky Normal Form

1. Chomsky Normal Form
   - consists of the following two types: $\begin{cases} A \to BC \\ A \to a \end{cases}$

(2) $A \to X_1X_2\ldots X_k$ $\begin{cases} X_i \in V \cup T \\ k \geqq 2 \end{cases}$

[Step 1] If X_i is a terminal a, make a new nonterminal X_i', and add

$A \to X_1X_2\ldots X_{i-1}X_i'X_{i+1}\ldots X_k$
$X_i' \to a$ OK

After applying the step to all X_i, relabel them, and we have…

(2') $A \to X_1X_2\ldots X_k$ $\begin{cases} X_i \in V \\ k \geqq 2 \end{cases}$ k=2 is also 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') $A \to X_1X_2\ldots X_k$ $\begin{cases} X_i \in V \\ k \geqq 3 \end{cases}$

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

$A \to X_1Y_1$
$Y_1 \to X_2Y_2,\ Y_2 \to X_3Y_3,\ \ldots,\ Y_{k-3} \to X_{k-2}Y_{k-2}$ OK
$Y_{k-2} \to X_{k-1}X_k$

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. Normal forms for CFL (Chomsky Normal Form)

A,B,C: Nonterminals
a: terminal
α : 0 or more nonterminals

7.1.5. Chomsky Normal Form

1. Chomsky Normal Form
   - consists of the following two types: $\begin{cases} A \to BC \\ A \to a \end{cases}$

(2') $A \to X_1X_2\ldots X_k$ $\begin{cases} X_i \in V \\ k \geqq 3 \end{cases}$

[Step 2] Make new nonterminals $Y_1,Y_2,\ldots,Y_{k-2}$, and rewrite

$A \to X_1Y_1$
$Y_1 \to X_2Y_2,\ Y_2 \to X_3Y_3,\ \ldots,\ Y_{k-3} \to X_{k-2}Y_{k-2}$ OK
$Y_{k-2} \to X_{k-1}X_k$

Now all rules are one of two types in Chomsky Normal Form.

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から実際に構成できる。

[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. Normal forms for CFL (Chomsky Normal Form)

A,B,C: Nonterminals
a: terminal
α : 0 or more nonterminals

7.1.5. Chomsky Normal Form

1. Chomsky Normal Form
   - consists of the following two types: $\begin{cases} A \to BC \\ A \to a \end{cases}$

[Summary] For any CFL L, there exists a CFG of Chomsky Normal Form that generates $L-\{\varepsilon\}$.

And we can surely construct from any G with $L(G)=L$.

[Note] For any CFL L, there exists a CFG of Greibach Normal Form that generates $L-\{\varepsilon\}$.
(Similar constructive proof is given.)