5. 文脈自由文法と言語(3): (テキスト5.4)

5.4. 文法と言語の曖昧さ

言語 L

文法 G

言語 = 文法的に正しい語の集合
• 各語は、文法による構造を持つ

CFLで言えば、複数の構文木を持つ

文法が曖昧
⇔ある語が文法上'正しい構造'を複数持つ

例) Time flies like an arrow.

「時は矢のように飛ぶ」のか?
「時蝿は矢を好む」のか?

5. Context Free Language (3): (Text 5.4)

5.4. Ambiguity of languages and grammars

Language L

Grammar G

Language = Set of grammatically correct words
• Each word has grammatical structure.

Grammar is ambiguous
⇔ ∃ word has two or more grammatical correct structures

In CFL, two or more parse trees.

Ex) Time flies like an arrow.

'Time' flies like an arrow?

'Time flies' like an arrow?

5. 文脈自由文法と言語(3): (テキスト5.4)

5.4. 文法と言語の曖昧さ

言語 L

文法 G

文法が曖昧
⇔ある語が文法上'正しい構造'を複数持つ

★適切な文法によって、曖昧さを回避できることがある。実用上、よく用いられる。

★CFL L で、「$L(G)=L$ を満たすどんな CFG G」も曖昧な文法になるものがある。つまり CFL は本質的な曖昧さをもつ。

5. 文脈自由文法と言語(3): (テキスト5.4)

5.4. Ambiguity of languages and grammars

Language L

Grammar G

Grammar is ambiguous
⇔ ∃ word has two or more grammatical correct structures

★Practically, ambiguity can be avoided by smart design of the grammar in most cases.

★There is a CFL L s. t. 'any CFG G with $L(G)=L$ is ambiguous.' That is, the class of CFL is essentially ambiguous.

5. 文脈自由文法と言語(3): (テキスト5.4)

5.4. 文法と言語の曖昧さ

– いくつかの違ったレベルの曖昧さ

1. 文法の構成を工夫すれば回避できる
   その言語に対して、上手に構成してやれば回避可能
2. 文法に付加的なルールを想定すれば回避できる
   文法＋αで回避可能
3. 本質的に曖昧
   言語 L を表現するどんな文法も曖昧になってしまう
   ⇒言語 L は本質的に曖昧である、と言う。

5. Context Free Language (3): (Text 5.4)

5.4. Ambiguity of languages and grammars

– Some different levels ambiguities

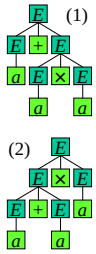
1. It can be avoided by some tricks for a construction of a grammar
   For some languages, we can avoid it if we are smart enough.
2. It can be avoided if we admit some additional rule out of grammar
   For some languages, we can avoid it by Grammar ＋α
3. It is inherently ambiguous
   For some languages, any grammar for the language is ambiguous.
   ⇒The language L is said to be inherently ambiguous.

5. 文脈自由文法と言語(3): (テキスト5.4)

5.4.1. 曖昧な文法

例) $G_1=(\{E\},\{+,\times,a\}, E\to E+E \mid E\times E \mid a, E)$

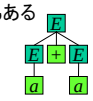
- 文形式 $a+a\times a$ の本質的に違う導出
  - (1) $E \Rightarrow E+E \Rightarrow E+E\times E \overset{*}{\Rightarrow} a+a\times a$
  - (2) $E \Rightarrow E\times E \Rightarrow E+E\times E \overset{*}{\Rightarrow} a+a\times a$



注) 導出が違っていても、本質的に同じ構造もある

$E \Rightarrow E+E \Rightarrow a+E \Rightarrow a+a$

$E \Rightarrow E+E \Rightarrow E+a \Rightarrow a+a$



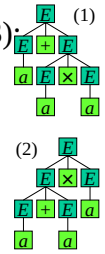
★本質的に違う導出＝構文木の形が違う

5. Context Free Language (3): (Text 5.4)

5.4.1. Ambiguity of grammars

Ex) $G_1=(\{E\},\{+,\times,a\}, E\to E+E \mid E\times E \mid a, E)$

- Sentential form $a+a\times a$ has two essentially different derivations
  - (1) $E \Rightarrow E+E \Rightarrow E+E\times E \overset{*}{\Rightarrow} a+a\times a$
  - (2) $E \Rightarrow E\times E \Rightarrow E+E\times E \overset{*}{\Rightarrow} a+a\times a$



Note) There is a same structure with different derivations

$E \Rightarrow E+E \Rightarrow a+E \Rightarrow a+a$

$E \Rightarrow E+E \Rightarrow E+a \Rightarrow a+a$

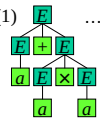
★essentially different derivations＝different parse trees

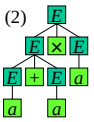
5. 文脈自由文法と言語(3): (テキスト5.4)

5.4.1. 曖昧な文法

例) $G_1=(\{E\},\{+,\times,a\}, E\to E+E \mid E\times E \mid a, E)$

- 文形式 $a+a\times a$ の本質的に違う導出
  - (1)$E \Rightarrow E+E \Rightarrow E+E\times E \overset{*}{\Rightarrow} a+a\times a$
  - (2) $E \Rightarrow E\times E \Rightarrow E+E\times E \overset{*}{\Rightarrow} a+a\times a$

(1)  …式 $a+(a\times a)$ に対応

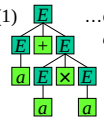
(2)  …式 $(a+a)\times a$ に対応

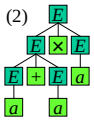
5. Context Free Language (3): (Text 5.4)

5.4.1. Ambiguity of grammars

Ex) $G_1=(\{E\},\{+,\times,a\}, E\to E+E \mid E\times E \mid a, E)$

- Sentential form $a+a\times a$ has two essentially different derivations
  - (1) $E \Rightarrow E+E \Rightarrow E+E\times E \overset{*}{\Rightarrow} a+a\times a$
  - (2) $E \Rightarrow E\times E \Rightarrow E+E\times E \overset{*}{\Rightarrow} a+a\times a$

(1)  …corresponds to $a+(a\times a)$

(2)  …corresponds to $(a+a)\times a$

5. 文脈自由文法と言語(3): (テキスト5.4)

5.4.1. 曖昧な文法

$G_1=(\{E\},\{+,\times,a\}, E\to E+E \mid E\times E \mid a, E)$ の曖昧さ

1. 演算子[+]と[×]の優先順位が表現できない
   - $(1)E \Rightarrow E+E \Rightarrow E+E\times E \overset{*}{\Rightarrow} a+\underline{a\times a}$ ←○
   - $(2)E \Rightarrow E\times E \Rightarrow E+E\times E \overset{*}{\Rightarrow} \underline{a+a}\times a$
2. 同じ演算子内の順番が不定
   - $(1)E \Rightarrow E+E \Rightarrow E+E+E \overset{*}{\Rightarrow} a+\underline{a+a}$
   - $(2)E \Rightarrow E+E \Rightarrow E+E+E \overset{*}{\Rightarrow} \underline{a+a}+a$ ←○
3. (導出の曖昧さ)

5. Context Free Language (3): (Text 5.4)

5.4.1. Ambiguity of grammars

Ambiguities of $G_1=(\{E\},\{+,\times,a\}, E\to E+E|E\times E|a, E)$

1. It cannot represent the priority between [+] and [×]
   - (1) $E \Rightarrow E+E \Rightarrow E+E\times E \overset{*}{\Rightarrow} a+\underline{a\times a}$ ←○
   - (2) $E \Rightarrow E\times E \Rightarrow E+E\times E \overset{*}{\Rightarrow} \underline{a+a}\times a$
2. The order among the same operators are not defined.
   - (1) $E \Rightarrow E+E \Rightarrow E+E+E \overset{*}{\Rightarrow} a+\underline{a+a}$
   - (2) $E \Rightarrow E+E \Rightarrow E+E+E \overset{*}{\Rightarrow} \underline{a+a}+a$ ←○

» (Ambiguity for derivations)

5. 文脈自由文法と言語(3): (テキスト5.4)

5.4.2. Remove 文法の曖昧さの除去

1. 演算子[+]と[×]の優先順位を表現する

「式」をもっと細かく定義しなおす;

1. 因数(*I*): 識別子 *a* または括弧で囲まれた式
2. 項(*F*): 因数の積、つまり因数を×でつないだもの
3. 式(*E*): 項の和、つまり項を + でつないだもの

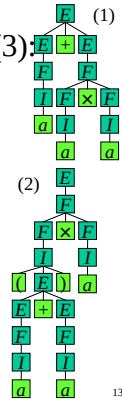
$G_2=(\{I,F,E\},\{+,\times,a,(,)\},A,E)$

$$A:\begin{cases} I \to a \mid (E) \\ F \to F\times F \mid I \\ E \to E+E \mid F \end{cases}$$

E⇒F⇒Iの順でしか展開できない

(1) $E \Rightarrow E+E \Rightarrow E+F \overset{*}{\Rightarrow} F+I\times I \overset{*}{\Rightarrow} a+a\times a$

(2) $E \Rightarrow F\Rightarrow F\times F \overset{*}{\Rightarrow} (E+E)\times F \overset{*}{\Rightarrow} (a+a)\times a$



5. Context Free Language (3): (Text 5.4)

5.4.2. Remove ambiguity of grammars

1. The priority between [+] and [×]

We grind a 'formula' down to more detailed concepts;

1. Identifier (*I*): Identifier *a* or a formula grouped by ().
2. Factor (*F*): Product of two identifiers
3. Expression (*E*): Sum of two factors

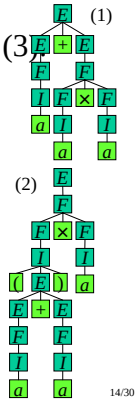
$G_2=(\{I,F,E\},\{+,\times,a,(,)\},A,E)$

$$A:\begin{cases} I \to a \mid (E) \\ F \to F\times F \mid I \\ E \to E+E \mid F \end{cases}$$

Expanded E⇒F⇒I

(1) $E \Rightarrow E+E \Rightarrow E+F \overset{*}{\Rightarrow} F+I\times I \overset{*}{\Rightarrow} a+a\times a$

(2) $E \Rightarrow F\Rightarrow F\times F \overset{*}{\Rightarrow} (E+E)\times F \overset{*}{\Rightarrow} (a+a)\times a$



5. 文脈自由文法と言語(3): (テキスト5.4)

5.4.2. 文法の曖昧さの除去

2. 同じ演算子内の順番を決める

[左を優先するための変数]を入れる: 例えば

$E \Rightarrow E+E \mid \alpha$

は以下の変形をする。

$F \Rightarrow \alpha$

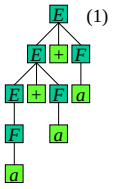
$E \Rightarrow E+F \mid F$

右辺のEが一つしかない

$G_3=(\{F,E\},\{+,\times,a\},A,E)$

$$A:\begin{cases} F \to a \\ E \to E+F \mid E\times F \mid F \end{cases}$$

(1) $E \Rightarrow E+F \Rightarrow E+F+F \Rightarrow F+F+F \overset{*}{\Rightarrow} a+a+a$



5. Context Free Language (3): (Text 5.4)

5.4.2. Remove ambiguity of grammars

2. Leftmost operator has priority

Use a variable meaning [priority for the leftmost operator]; for example,

$E \Rightarrow E+E \mid \alpha$

is rewritten with *F* as

$F \Rightarrow \alpha$

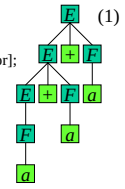
$E \Rightarrow E+F \mid F$

The rule only has one *E* on right.

$G_3=(\{F,E\},\{+,\times,a\},A,E)$

$$A:\begin{cases} F \to a \\ E \to E+F \mid E\times F \mid F \end{cases}$$

(1) $E \Rightarrow E+F \Rightarrow E+F+F \Rightarrow F+F+F \overset{*}{\Rightarrow} a+a+a$



5. 文脈自由文法と言語(3): (テキスト5.4)

5.4.2. 文法の曖昧さの除法

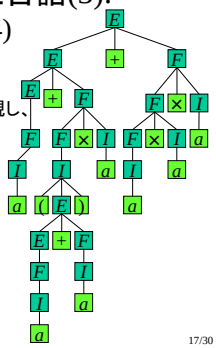
1. 演算子[+]と[×]の優先順位を表現し、
2. 左優先を表現する

$G_4=(\{I,F,E\},\{+,\times,a,(,)\},A,E)$

$$A:\begin{cases} I \to a \mid (E) \\ F \to F\times I \mid I \\ E \to E+F \mid F \end{cases}$$

例) $E \overset{*}{\Rightarrow} a+(a+a)\times a+a\times a\times a$

に対する構文木は右のものだけ



5. Context Free Language (3): (Text 5.4)

5.4.2. Remove ambiguity of grammars

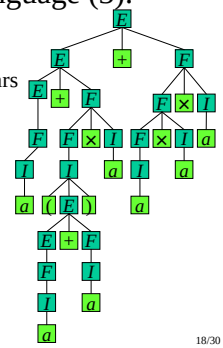
1. Priority between [+] and [×]
2. Priority for the leftmost operator

$G_4=(\{I,F,E\},\{+,\times,a,(,)\},A,E)$

$$A:\begin{cases} I \to a \mid (E) \\ F \to F\times I \mid I \\ E \to E+F \mid F \end{cases}$$

Ex) The unique parse tree for

$E \overset{*}{\Rightarrow} a+(a+a)\times a+a\times a\times a$



5.4.3. 曖昧さを表現する手段としての最左導出

3. 導出に関する曖昧さをなくす
…最左導出によって、構文木の「たどり方」は一意的に決まる。

[定理] 語の構文木の個数と最左導出の個数は同じ

1,2 の対策によって、与えられた「式」の構文木は一意的に決まる。
3 の対策によって、導出の順番に関する曖昧さはなくなる。

実用上、多くのCFLは上記の方法で
曖昧でないCFGを構築することができる。

5.4.3. Ambiguity of derivations

3. The ambiguity of derivations
…Using the leftmost derivation, the ambiguity of derivations is avoided; the way to sweep the parse tree is uniquely determined.

[Theorem] For a word, the number of parse trees equals to the number of the leftmost derivations.

By two tricks 1 and 2, the parse tree for a formula is uniquely determined.
By the leftmost derivation, the ambiguity of derivations is avoided.

Practically, we can construct
unambiguous CFG for most CFL.

5. 文脈自由文法と言語(3): (テキスト5.4)

5.4. 文法と言語の曖昧さ

5.4.4. 本質的な曖昧さ

言語 L が本質的に曖昧 ⇔ L を表現する任意の文法が曖昧

- ✓ CFLは本質的に曖昧な言語を含む
- ✓ 与えられた言語が本質的に曖昧かどうかは、計算によって判定することはできない

⇒ここでは本質的に曖昧な言語の例を示すにとどめる

5. Context Free Language (3): (Text 5.4)

5.4. Ambiguity of languages and grammars

5.4.4. Inherent ambiguity

A language L is inherently ambiguous ⇔ Any grammar G with $L(G)=L$ is ambiguous.

- ✓ CFL contains inherently ambiguous languages.
- ✓ For some language L, we cannot compute if L is inherently ambiguous or not.

⇒We only show an example language that is inherently ambiguous.

5. 文脈自由文法と言語(3): (テキスト5.4)

5.4. 文法と言語の曖昧さ

5.4.4. 本質的な曖昧さ

言語 $L = \{ a^n b^n c^m d^m \mid n \geqq 1, m \geqq 1\}$
$\cup \{a^n b^m c^m d^n \mid n \geqq 1, m \geqq 1\}$

L は、$aa^*bb^*cc^*dd^*$ の部分集合で、

- a と b が同数かつ c と d が同数、または
- a と d が同数かつ b と c が同数

であるような語の集合

5. Context Free Language (3): (Text 5.4)

5.4. Ambiguity of languages and grammars

5.4.4. Inherent ambiguity

Language $L = \{ a^n b^n c^m d^m \mid n \geqq 1, m \geqq 1\}$
$\cup \{a^n b^m c^m d^n \mid n \geqq 1, m \geqq 1\}$

L is a subset of $aa^*bb^*cc^*dd^*$ such that

- a and b are the same number, and so are c and d, or
- a and d are the same number, and so are b and c.

5. 文脈自由文法と言語(3): (テキスト5.4)

5.4. 文法と言語の曖昧さ

5.4.4. 本質的な曖昧さ

言語 $L=\{a^n b^n c^m d^m\} \cup \{a^n b^m c^m d^n\}$ を表現する文法の例:

$G=(\{S,A,B,C,D\},\{a,b,c,d\},X,S)$

X:
$S \rightarrow AB \mid C$
$A \rightarrow aAb \mid ab$
$B \rightarrow cBd \mid cd$
$C \rightarrow aCd \mid aDd$
$D \rightarrow bDc \mid bc$

前者/後者の別

語 $a^k b^k c^k d^k$ は???

5. Context Free Language (3): (Text 5.4)

5.4. Ambiguity of languages and grammars

5.4.4. Inherent ambiguity

A grammar that represents $L=\{a^n b^n c^m d^m\} \cup \{a^n b^m c^m d^n\}$:

$G=(\{S,A,B,C,D\},\{a,b,c,d\},X,S)$

X:
$S \rightarrow AB \mid C$
$A \rightarrow aAb \mid ab$
$B \rightarrow cBd \mid cd$
$C \rightarrow aCd \mid aDd$
$D \rightarrow bDc \mid bc$

Former / Latter

What if $a^k b^k c^k d^k$???

5. 文脈自由文法と言語(3): (テキスト5.4)

5.4. 文法と言語の曖昧さ

5.4.4. 本質的な曖昧さ

[定理] 言語 $L=\{a^n b^n c^m d^m\} \cup \{a^n b^m c^m d^n\}$ は本質的に曖昧である

[証明のアイデア]

L を表現するどんな文法も語 $a^k b^k c^k d^k$ に対しては複数の構文木を持つことを示す。$a^n b^n c^m d^m$ を生成する規則と、$a^n b^m c^m d^n$ を生成する規則の両方とも $a^k b^k c^k d^k$ を生成せざるをえない。

5. Context Free Language (3): (Text 5.4)

5.4. Ambiguity of languages and grammars

5.4.4. Inherent ambiguity

[Theorem] $L=\{a^n b^n c^m d^m\} \cup \{a^n b^m c^m d^n\}$ is inherently ambiguous.

[Proof idea]

Any grammar G with $L(G)=L$ has at least two parse trees for the word $a^k b^k c^k d^k$, since $a^k b^k c^k d^k$ has to be derived by the rules for $a^n b^n c^m d^m$ and the rules for $a^n b^m c^m d^n$.

5.4. 文法と言語の曖昧さ

5.4.4. 本質的な曖昧さ

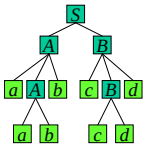
言語 $L=\{a^n b^n c^m d^m\} \cup \{a^n b^m c^m d^n\}$ を表現する文法の例:

$G=(\{S,A,B,C,D\},\{a,b,c,d\},X,S)$

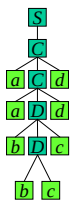
X: $S \rightarrow AB \mid C$; $A \rightarrow aAb \mid ab$; $B \rightarrow cBd \mid cd$; $C \rightarrow aCd \mid aDd$; $D \rightarrow bDc \mid bc$

語 $aabbccdd$ の2つの構文木

$a^n b^n c^m d^m$ の要素と見たとき



$a^n b^m c^m d^n$ の要素と見たとき



5.4. Ambiguity of languages and grammars

5.4.4. Inherent ambiguity

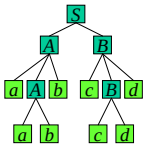
A grammar that represents $L=\{a^n b^n c^m d^m\} \cup \{a^n b^m c^m d^n\}$:

$G=(\{S,A,B,C,D\},\{a,b,c,d\},X,S)$

X: $S \rightarrow AB \mid C$; $A \rightarrow aAb \mid ab$; $B \rightarrow cBd \mid cd$; $C \rightarrow aCd \mid aDd$; $D \rightarrow bDc \mid bc$

Two parse trees for the word $aabbccdd$

As an element in $a^n b^n c^m d^m$



As an element in $a^n b^m c^m d^n$