5. 文脈自由文法と言語(2):
(テキスト5.2)

5.2.3. 推論･導出･構文木

1→5, 5→3, 2→1 を示す。

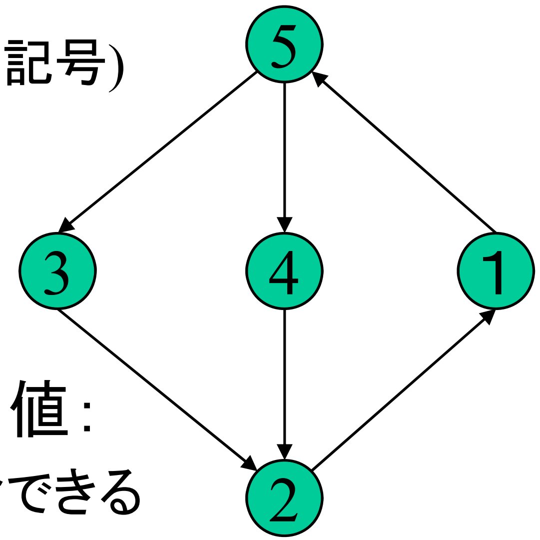
- 再帰的推論

  文字列(語＝終端記号列)から出発記号(非終端記号)

– 導出(最左導出と最右導出)

  出発記号(非終端記号)から文字列(語)

– 構文木



文法 $G=(V,T,P,S)$ について以下はすべて同値:

1. 終端記号列 w から変数 S が再帰的に推論できる
2. $S \overset{*}{\Rightarrow} w$
3. $S \underset{左}{\overset{*}{\Rightarrow}} w$
4. $S \underset{右}{\overset{*}{\Rightarrow}} w$
5. S を根とし、w を成果とする構文木が存在。

- 3→2, 4→2 は自明
- 5→3, 5→4 は対称

5. 文脈自由文法と言語(2):
(テキスト5.2)

5.2.4. 再帰的推論から構文木へ(1→5)

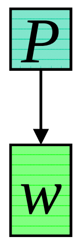
[定理] CFG G=(V,T,P,S) に対し、再帰的推論で語 w が変数 S の言語に属しているなら、S を根として、w を成果とする構文木が存在する。

[証明] w が S の言語に属していることを示す導出のステップ数に関する帰納法。

[基礎] w が S から1ステップで導出できる場合

生成規則 S→w が P に入っている。

したがって構文木が存在。



5.2.4. 再帰的推論から構文木へ(1→5)

[定理] CFG $G=(V,T,P,S)$ に対し、再帰的推論で語 w が変数 S の言語に属しているなら、S を根として、w を成果とする構文木が存在する。

[証明] w が S の言語に属していることを示す導出のステップ数に関する帰納法。

[帰納] w が S から n+1 ステップ(n＞1)で導出できる場合

[帰納法の仮定] G において語 x が変数 B の言語に属していて、かつ x が B から n ステップ以下で導出できるなら、B を根として、x を成果とする構文木が存在。

5.2.4. 再帰的推論から構文木へ(1→5)

[証明] w が S の言語に属していることを示す導出のステップ数に関する帰納法。

[帰納] w が S から n+1 ステップ(n＞1)で導出できる場合

[帰納法の仮定] G において語 x が変数 B の言語に属していて、かつ x が B から n ステップ以下で導出できるなら、B を根として、x を成果とする構文木が存在。

w は S から n+1 ステップで導出できるので、P は生成規則

$$S \to X_1 X_2 \ldots X_k$$

をもち、かつ

$$X_i \overset{*}{\Rightarrow} w_i$$

$$w = w_1 w_2 \ldots w_{\mathrm{k}}$$

を満たす文字列 $w_1, w_2, \ldots, w_k$ が存在する。

X_iからw_iの導出は高々 n ステップ
($X_i = w_i$ もありえる)

5.2.4. 再帰的推論から構文木へ(1→5)

[帰納] w が S から n+1 ステップ(n＞1)で導出できる場合

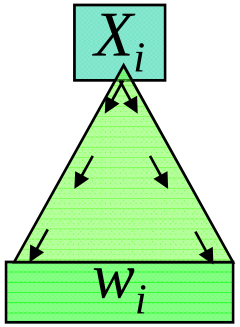
[帰納法の仮定] G において語 x が変数 B の言語に属していて、かつ x が B から n ステップ以下で導出できるなら、B を根として、x を成果とする構文木が存在。

w は S から n+1 ステップで導出できるので、P は規則

$S \rightarrow X_1X_2\ldots X_k$ をもち、かつ

$X_i \overset{*}{\Rightarrow} w_i$ (nステップ以下で導出できる)

$w = w_1w_2\ldots w_\mathrm{k}$



を満たす文字列 $w_1, w_2, \ldots, w_k$ が存在する。

G において語 w_i は変数 X_i の言語に属し、かつ n ステップ以下で導出できるので、帰納法の仮定より、X_i を根として w_i を成果とする構文木が存在する。

5.2.4. 再帰的推論から構文木へ(1→5)

[帰納] w が S から n+1 ステップ(n＞1)で導出できる場合

w は S から n+1 ステップで導出できるので、P は生成規則

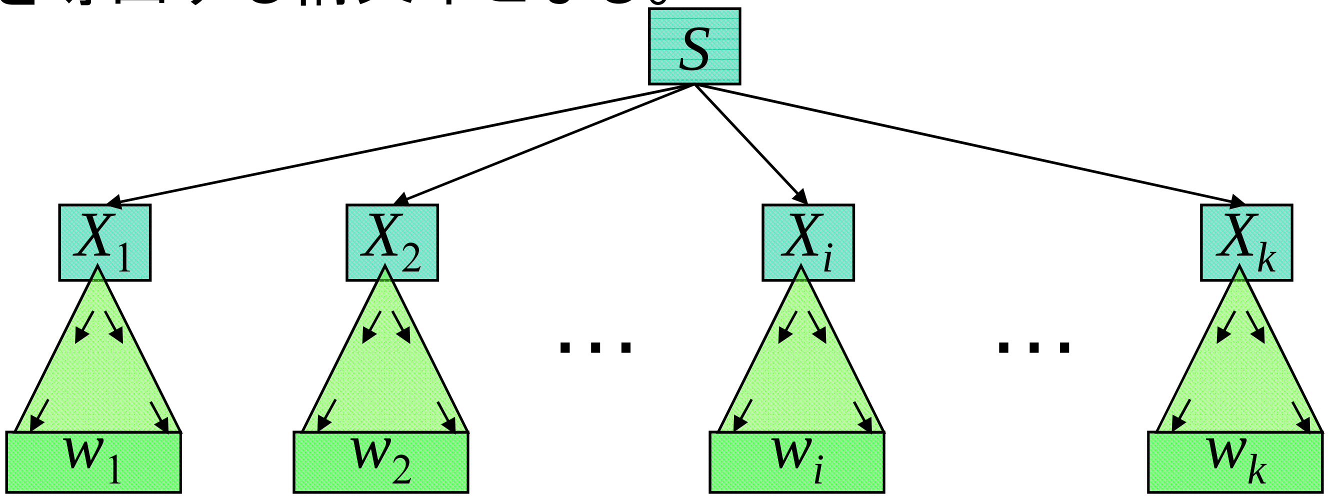
$$S \rightarrow X_1 X_2 \ldots X_k$$

をもち、かつ

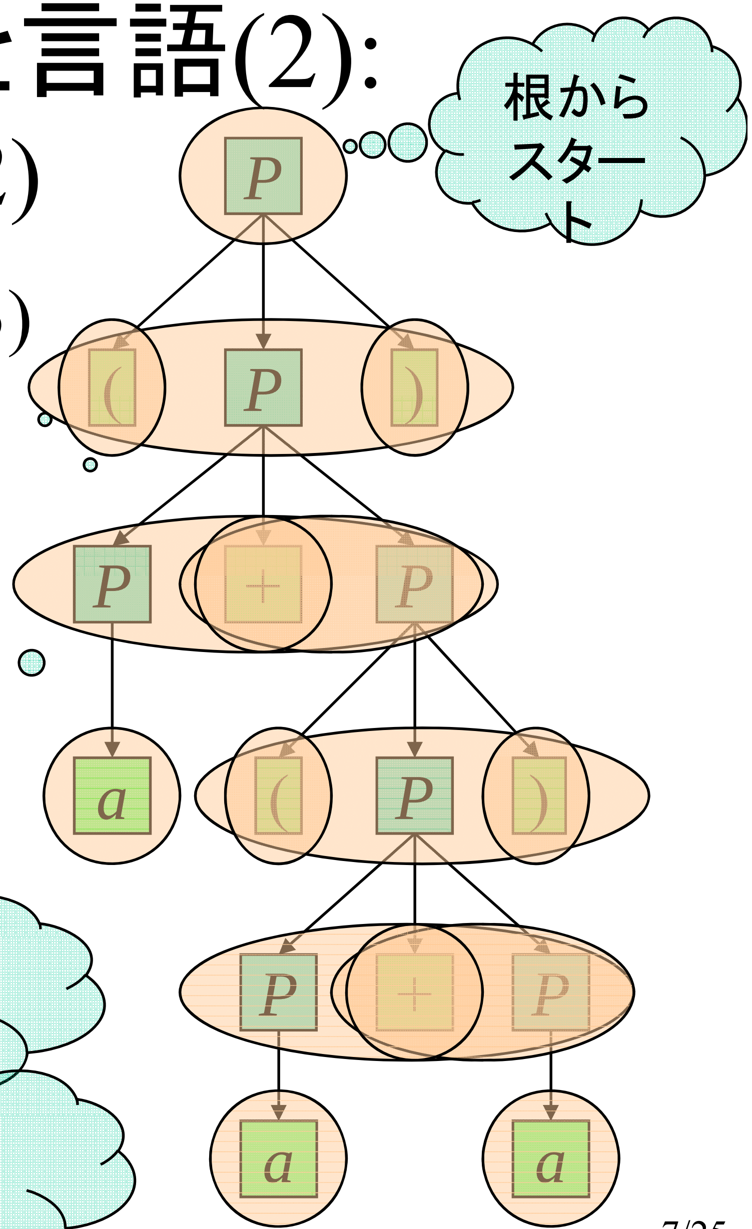
$$X_i \overset{*}{\Rightarrow} w_i, \; w = w_1 w_2 \ldots w_{\mathrm{k}}$$

を満たす文字列 $w_1, w_2, \ldots, w_k$ が存在する。

帰納法の仮定より、X_i を根として w_i を成果とする構文木が存在する。これらの構文木から以下の構文木を構成すると、S から w を導出する構文木となる。



5. 文脈自由文法と言語(2): (テキスト5.2)



5.2.5. 構文木から最左導出へ(5→3)

直感的には…

構文木を左優先で
探索を行う
ことに対応する。

例) $P \Rightarrow (P) \Rightarrow (P+P)$

$\Rightarrow (a+P) \Rightarrow (a+(P))$

$\Rightarrow a+(P+P))$

$\Rightarrow (a+(a+P))$

$\Rightarrow (a+(a+a))$

終端記号(葉)
はそこで終わ
り

非終端記号
は左優先で

5. 文脈自由文法と言語(2): (テキスト5.2)

5.2.5. 構文木から最左導出へ(5→3)

[定理] CFG $G=(V,T,P,S)$ に対し、変数 S を根とし、w を成果とする構文木があれば、G の最左導出 $S \overset{*}{\underset{左}{\Rightarrow}} w$ が存在する。

[略証] 木の高さ i についての帰納法で証明する。

(木の<u>高さ</u> = 各葉から根までの距離の最大値)

木の高さが0のときは根しかないので、これは構文木ではない。したがって意味のある木の高さの最小値は1。

[基礎] $i=1$のとき: $S \to w$ が規則に入っている。

これは最左導出。

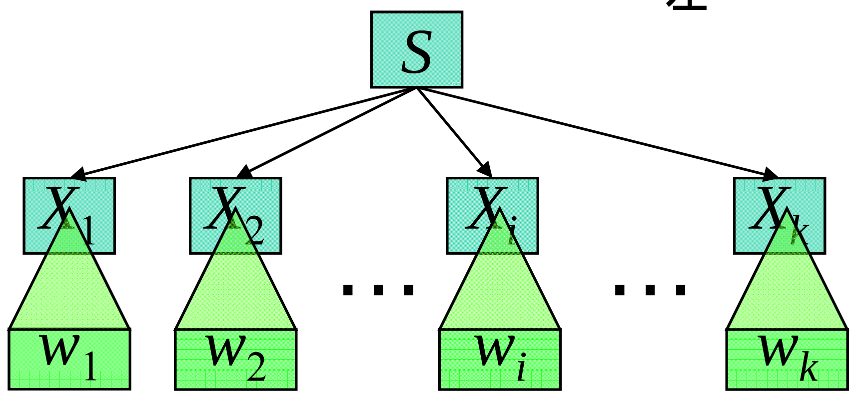
5.2.5. 構文木から最左導出へ(5→3)

[定理] CFG $G=(V,T,P,S)$ に対し、変数 S を根とし、w を成果とする構文木があれば、G の最左導出 $S \overset{*}{\underset{左}{\Rightarrow}} w$ が存在する。

[略証] 木の高さ i についての帰納法で証明する。

[帰納] $i \geqq 2$以上のとき:

- 根のラベルを S とし、S の子供のラベルを左から $X_1, X_2, \ldots, X_k$ とする。
- 帰納法の仮定から、各 X_i の成果 w_i に対して、最左導出 $X_i \overset{*}{\underset{左}{\Rightarrow}} w_i$ が存在する。
- $w = w_1 w_2 \ldots w_k$ である。



導出 $S \underset{左}{\Rightarrow} X_1 X_2 \ldots X_k$ から、最左導出

$$S \overset{*}{\underset{左}{\Rightarrow}} w_1 w_2 \ldots w_k$$

が構成できることを示す。具体的には $j=1,2,\ldots,k$ について、

$$S \overset{*}{\underset{左}{\Rightarrow}} w_1 w_2 \ldots w_j X_{j+1} \ldots X_k$$

であることを j に関する帰納法で示す。(以下省略)

5. 文脈自由文法と言語(2):
(テキスト5.2)

5.2.6. 導出から再帰的推論へ(2→1)

[定理] CFG $G=(V,T,P,S)$ に対して、導出 $S\overset{*}{\Rightarrow}w$ があれば、w が S の言語に属することが再帰的推論によって確かめられる。

[略証] 導出 $S\overset{*}{\Rightarrow}w$ の長さに関する帰納法による。

[基礎] 長さが 1 のとき: $S\rightarrow w$ が生成規則に入っている。

したがって1ステップの再帰的推論により確認できる。

[帰納] 導出 $S\overset{*}{\Rightarrow}w$ の長さが $n+1$ とし、長さ n 以下のすべての導出が再帰的推論によって確かめられるとする。

導出は生成規則 $S\rightarrow X_1X_2\ldots X_k$ により

$$S\Rightarrow X_1X_2\ldots X_k\overset{*}{\Rightarrow}w$$

という形で表現できる。

5.2.6. 導出から再帰的推論へ(2→1)

[定理] CFG $G=(V,T,P,S)$ に対して、導出 $S\overset{*}{\Rightarrow}w$ があれば、w が S の言語に属することが再帰的推論によって確かめられる。

[略証] 導出 $S\overset{*}{\Rightarrow}w$ の長さに関する帰納法による。

[帰納] 導出 $S\overset{*}{\Rightarrow}w$ の長さが $n+1$ とし、長さ n 以下のすべての導出が再帰的推論によって確かめられるとする。

導出は生成規則 $S\rightarrow X_1X_2\ldots X_k$ により

$$S\Rightarrow X_1X_2\ldots X_k\overset{*}{\Rightarrow}w$$

という形で表現できる。さらに

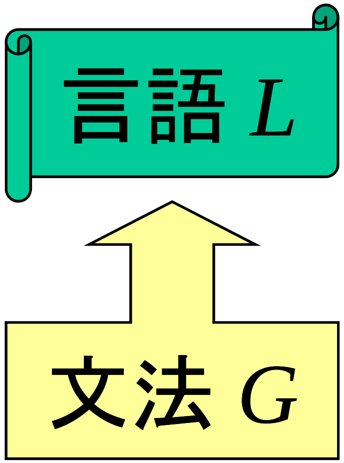
» $X_i\overset{*}{\Rightarrow}w_i \quad (1\leqq i\leqq k)$

» $w=w_1w_2\ldots w_k$

であり、帰納法の仮定から、すべての導出 $X_i\overset{*}{\Rightarrow}w_i$ は再帰的推論によって確かめられる。したがって $S\rightarrow X_1X_2\ldots X_k$ とこれらの推論から、w が S の言語に属することが推論によって確かめられる。

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 = 文法的に正しい語の集合
- 各語は、文法による<u>構造</u>を持つ

CFLで言えば、<u>複数の構文木を持つ</u>

<u>文法が曖昧</u>
⇔ある語が文法上‘正しい構造’を複数持つ

言語によっては文法の曖昧さを除くことができる。しかし、本質的に曖昧な言語もある。

★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。

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

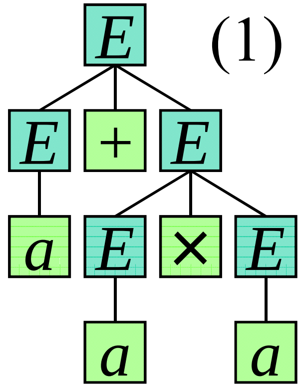
3. 本質的に曖昧

   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. 文脈自由文法と言語(3): (テキスト5.4)

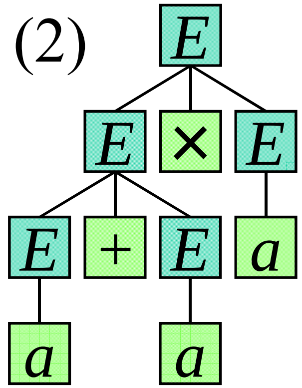
(1)



5.4.1. 曖昧な文法

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

(2)



- 文形式 $a+a\times a$ の本質的に違う導出

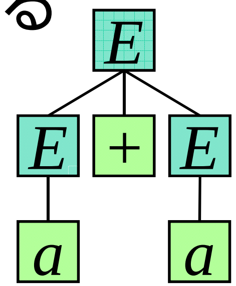
(1) $E \Rightarrow E+E \Rightarrow E+E\times E \overset{*}{\Rightarrow} a+a\times a$

(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

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

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



★<u>本質的に違う導出</u>＝<u>導出木の形</u>が違う

5. 文脈自由文法と言語(3): (テキスト5.4)

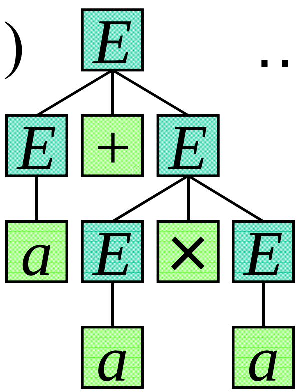
5.4.1. 曖昧な文法

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

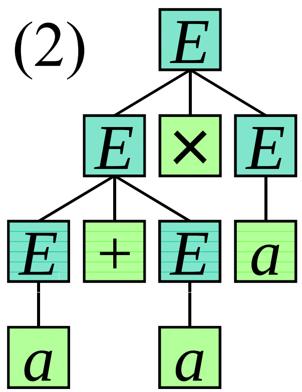
- 文形式 $a+a\times a$ の本質的に違う導出

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

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

(1) 

…式 $a+(a\times a)$ に対応

(2) 

…式 $(a+a)\times a$ に対応

5. 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=$(\{E\},\{+,\times,a\}, E\rightarrow E+E \mid E\times E \mid a, E)$ 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. 文脈自由文法と言語(3): (テキスト5.4)

$E \Rightarrow F \Rightarrow I$の順でしか展開できない

5.4.2. 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;

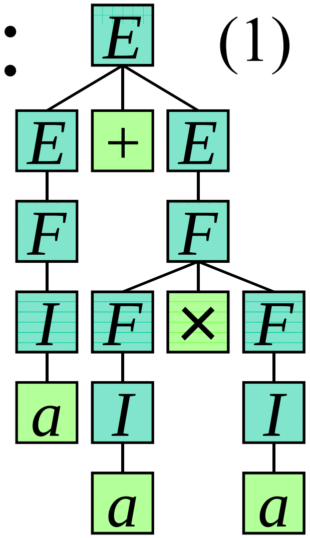
1. 因数(I): 識別子 a または括弧で囲まれた式
2. 項(F): 因数の積、つまり因数を×でつないだもの
3. 式(E): 項の和、つまり項を + でつないだもの

$$G_2=(\{I,F,E\},\{+,\times,a,(,)\}, A, E) \qquad A:\begin{cases} I \to a \mid (E) \\ F \to F \times F \mid I \\ E \to E + E \mid F \end{cases}$$

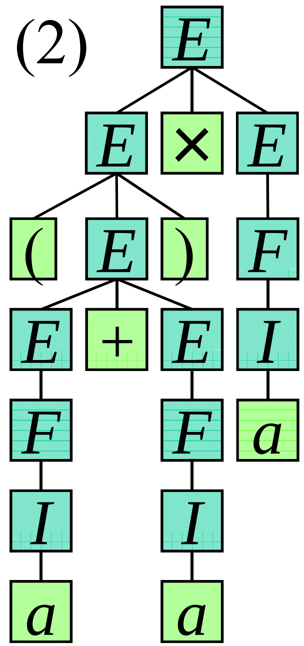
$$(1)\ E \Rightarrow E+E \Rightarrow E+F \overset{*}{\Rightarrow} F+I \times I \overset{*}{\Rightarrow} a+a \times a$$

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

(1)



(2)



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。

$F \Rightarrow \alpha$

$E \Rightarrow E+F \mid F$

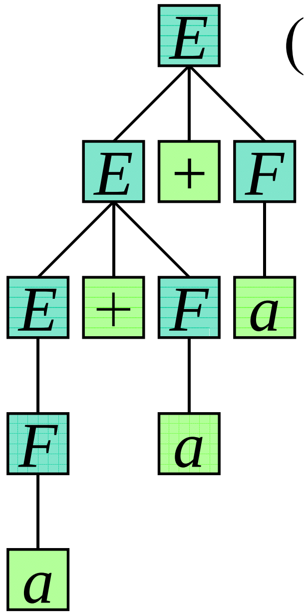
左にしか伸展できない

G_3=({F,E},{+,×,a}, A, E)

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

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

(1)



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

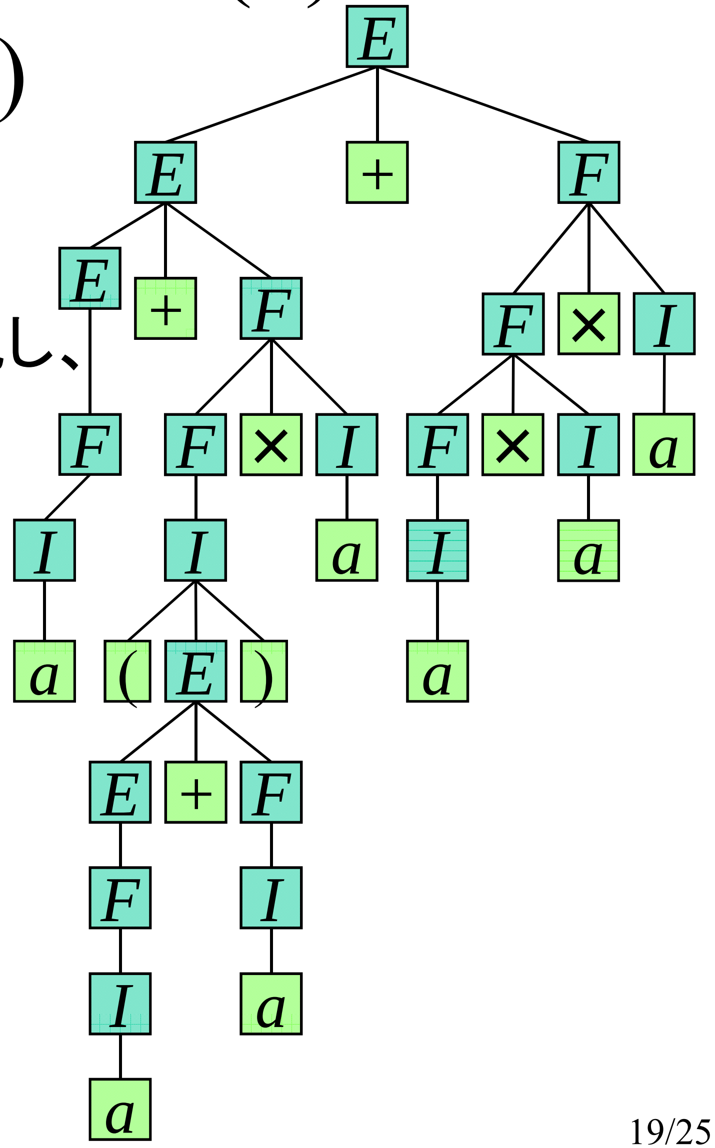
1. 演算子[+]と[×]の優先順位を表現し、
2. 左優先を表現する

G_4=({I,F,E},{+,×,a,(,)}, A, E)

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

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

に対する導出木は右のものだけ



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、導出木の「たどり方」は一意的に決まる。

[定理] 語の導出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」の導出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. 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 $\Leftrightarrow$ 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. 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、$a^+b^+c^+d^+$(または$aa^*bb^*cc^*dd^*$) 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. 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:

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

$$X: \begin{cases} S \to AB \mid C \\ A \to aAb \mid ab \\ B \to cBd \mid cd \\ C \to aCd \mid aDd \\ D \to bDc \mid bc \end{cases}$$

前者/後者の別

語 $a^k b^k c^k d^k$ は???

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 $a^k b^k c^k d^k$ に対しては複数の導出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。

5.4. 文法と言語の曖昧さ

5.4.4. 本質的な曖昧さ

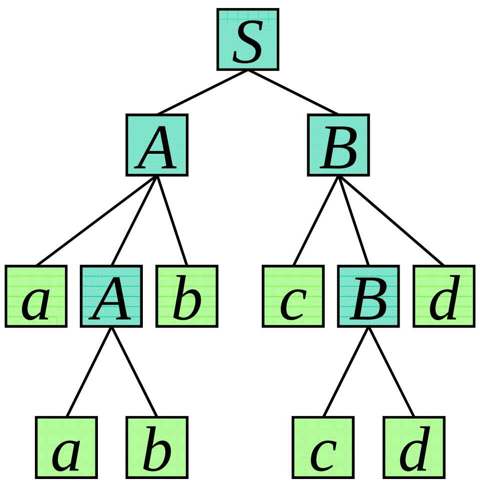
言語 $L=\{a^n b^n c^m d^m\} \cup \{a^n b^m c^m d^n\}$ を表現する文法の例:

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

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

語 *aabbccdd* の2つの導出木

$a^n b^n c^m d^m$ の要素と見たとき



$a^n b^m c^m d^n$ の要素と見たとき