句構造文法

概念の整理のための、いくつかの定義

定義1

- アルファベット(語彙):記号の有限集合
- 文:記号の(重複を許した)任意の有限列
- 言語:一つの「アルファベット」上の「文」の、任意の集合
- 手続き(procedure):実行可能な命令の有限列
- アルゴリズム:停止する「手続き」

- 構文(syntax)
  - 正しい「文」(well-formed sentence)を定義するための規則の集合P
  - 記号の並びについての議論。「意味」と独立
- 意味(semantics)
  - 記号、その系列と意味要素についての関係。
- 研究対象
  - 文の生成 .... 文法によって「文」が生成されるさま
  - 文法チェック .... 記号列が文法にあっているか
  - 構文解析 .... 「文」の句構造を探る
- 句構造文法
  - 1956年 チョムスキー(Noam Chomsky)による

- オートマトン(Automata)
  - 認識機械として:

    記号列 → 言語か？（YES, NO）
  - 変換機械として:

    入力記号列 → 出力記号列

よく使う記法

V^+：１個以上のVの要素の系列

例）

$V^+ = V \cup VV \cup VVV \cup \ldots = \bigcup_{i=1}^{\infty} V^i$

$V^* = \{\lambda\} \cup V^+$

空文字列

$\Sigma = \{a, b, c\}$　$V_N = \{A, B, C\}$

のとき、

$V = \Sigma \cup V_N = \{a, b, c, A, B, C\}$

$a, ab, aac, abcd, \ldots \in \Sigma^+$

$\lambda, A, a, AaC, AAaa, \ldots \in V^*$

句構造文法の定義

定義2：句構造文法　　$G = (V, \Sigma, P, S)$

- V:記号の集合
- Σ(⊂V)：アルファベット
  要素は終端記号(terminal symbol)という。
- $$V_N = V - \Sigma$$
  要素は非終端記号(nonterminal symbol)
  $$V = \sum \cup V_N \qquad \sum \cap V_N = \phi$$
- P:ルール（構文規則）の集合
  要素はα→β,　ただし　$\alpha \in V^+, \beta \in V^*$
- S:開始記号(initial symbol)

$$S \in V_N$$

例 Ga=(Va,Σa,Pa,S)

Σa={goto, if, read, I, B, L1, L2}

Va={<行き先>, <条件>, <名札>, <ます目>,S}

∪ Σa

Pa={S→<行き先><条件>,

<行き先> → goto <名札>,

<条件> → if read <ます目>,

<名札> → L1, <名札> → L2,

<ます目> → I, <ます目> → B }

Ga から生成される文

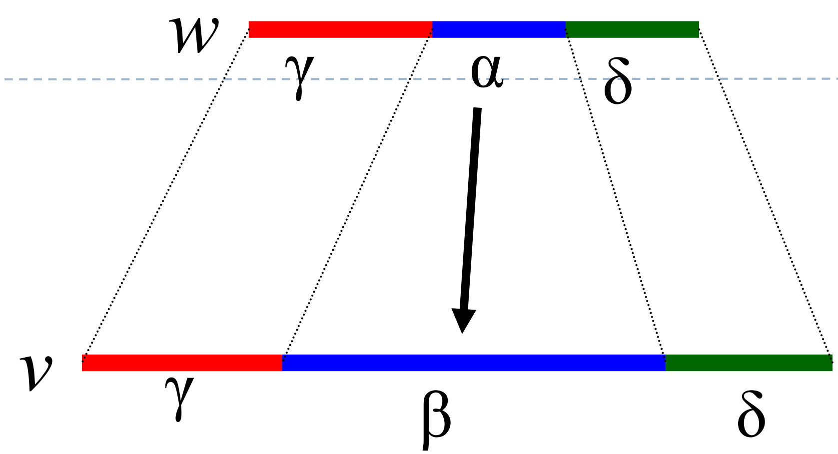
goto L1 if read B

goto L1 if read 1

goto L2 if read B

goto L2 if read 1

導出の定義



定義3:

$$w \Rightarrow v$$

iff

$$w = \gamma\alpha\delta, v = \gamma\beta\delta, \alpha \to \beta \in P, \gamma, \delta \in V^*$$

定義4:

$$w \overset{*}{\Rightarrow} v$$

wはvを生成する

iff

$$w = w_1 \Rightarrow w_2 \Rightarrow w_3 \Rightarrow \ldots \Rightarrow w_{n-1} \Rightarrow w_n = v$$

$$w_i \in V^*$$

言語の定義

定義5:　　$G = (V, \Sigma, P, S)$

文法Gによって生成される言語

$$L(G) = \{ w \mid w \in \Sigma^*, S \overset{*}{\Rightarrow} w \}$$

$$S \overset{*}{\Rightarrow} a \qquad \begin{cases} a \in \mathrm{V}^* & \text{のとき}a\text{は文形式という} \\ a \in \Sigma^* & \text{のとき}a\text{は文という} \end{cases}$$

文法に４つの型(ルールに制限)

- タイプ0文法G0： 制限なし

$$G = (V, \Sigma, P, S)$$

$$\alpha \to \beta \in \mathrm{P}, \alpha \in \mathrm{V}^{+},\ \beta \in \mathrm{V}^{*}$$

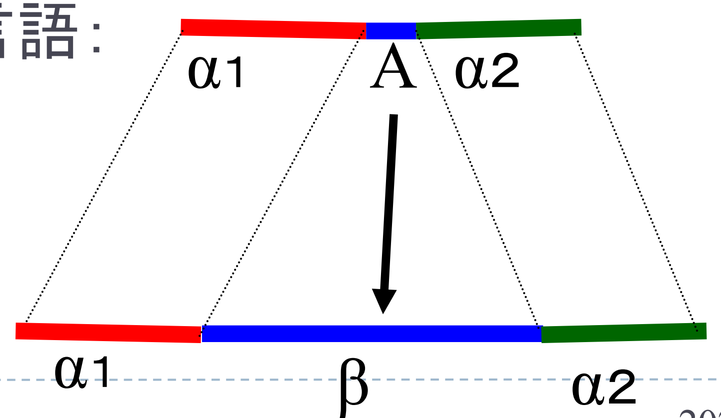
- タイプ1文法GI: 文脈依存文法(csg)

$$\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2 \in \mathrm{P}$$

$$\alpha_1, \alpha_2 \in \mathrm{V}^{*},\ \beta \in \mathrm{V}^{+}, A \in V_N$$

  - 前後にα1,α2を持つAをβに書きかえてよい
  - タイプ1文法から生成される言語：

  ⇒文脈依存言語(csl)



タイプ2文法G2:文脈自由文法（cfg）

$$A \to \beta \ \in \ \mathrm{P} \qquad \beta \in \mathrm{V}^{+}, A \in V_N$$

- Aをいつでもβに置き換えていい
- タイプ2文法から生成される言語:

⇒ 文脈自由言語(cfl)

タイプ3文法G3:正規文法(rg)

$$A \to a \in P, A \to aB \in P \qquad a \in \sum,\ A, B \in V_N$$

- タイプ3文法から生成される言語:

⇒ 正規言語（rl）

言語のクラスの包含関係

定理１：タイプ１であるが、タイプ２でない言語

がある。 $L(G_c) = \{a^n b^n c^n \mid n \geq 1\}$

定理２：タイプ２であるが、タイプ３でない言語

がある。 $L(G_b) = \{a^n b^n \mid n \geq 1\}$

定理１，２の証明としての反例

例） $G_b = (\{S,a,b\},\{a,b\},\{S \to aSb, S \to ab\}, S)$

$L(G_b) = \{a^n b^n \mid n \geq 1\}$

例） $G_c = (\{S,B,C,a,b,c\},\{a,b,c\}, P_c, S)$

$$P_c = \{S \to aSBC, S \to aBC,$$
$$CB \to BC,$$
$$aB \to ab, bB \to bb,$$
$$bC \to bc, cC \to cc\}$$

$S \stackrel{*}{\Rightarrow} a^n (BC)^n$

$\stackrel{*}{\Rightarrow} a^n B^n C^n$

$\stackrel{*}{\Rightarrow} a^n b^n C^n$

$\stackrel{*}{\Rightarrow} a^n b^n c^n$

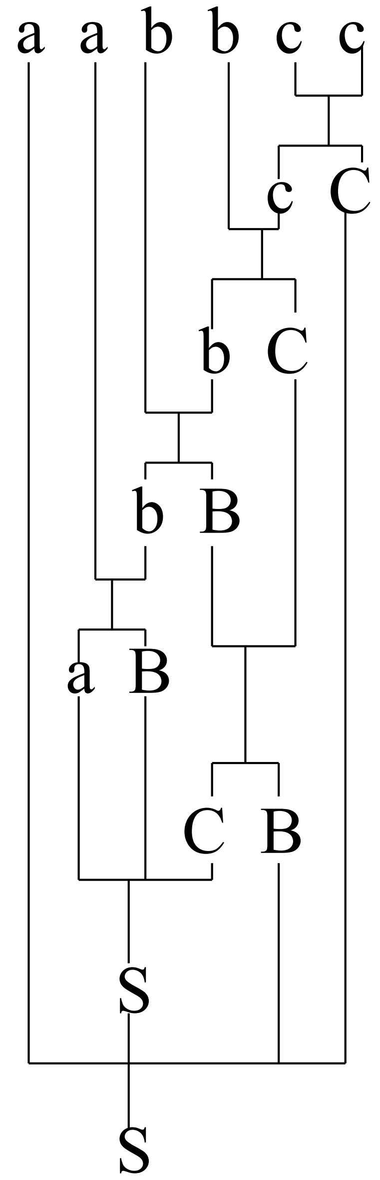
タイプ１で記述可能

$\{\mathrm{CB} \to \mathrm{CB}_1,$
$\mathrm{CB}_1 \to \mathrm{BB}_1,$
$\mathrm{BB}_1 \to \mathrm{BC}\}$

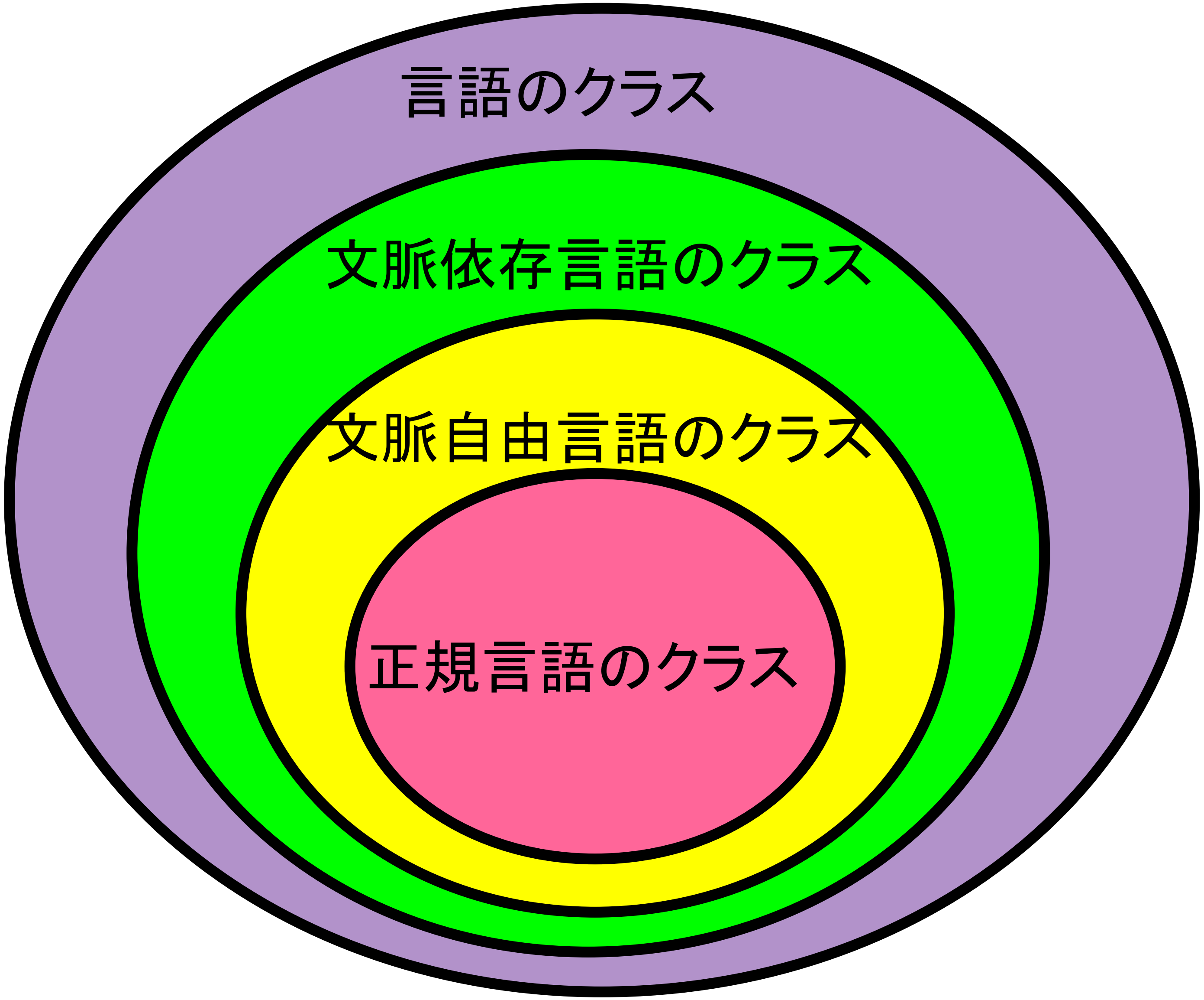
$L(G_c) = \{a^n b^n c^n \mid n \geq 1\}$

導出の例

aabbcc	導出で使われるルール
<= aabbcC	1.cC→cc
<=aabbCC	2.bC→bc
<=aabBCC	3.bB→bb
<=aaBBCC	4.aB→ab
<=aaBCBC	5.S→aSBC
<=aSBC	6.S→aBC
<=S	7.CB→BC



言語の包含関係

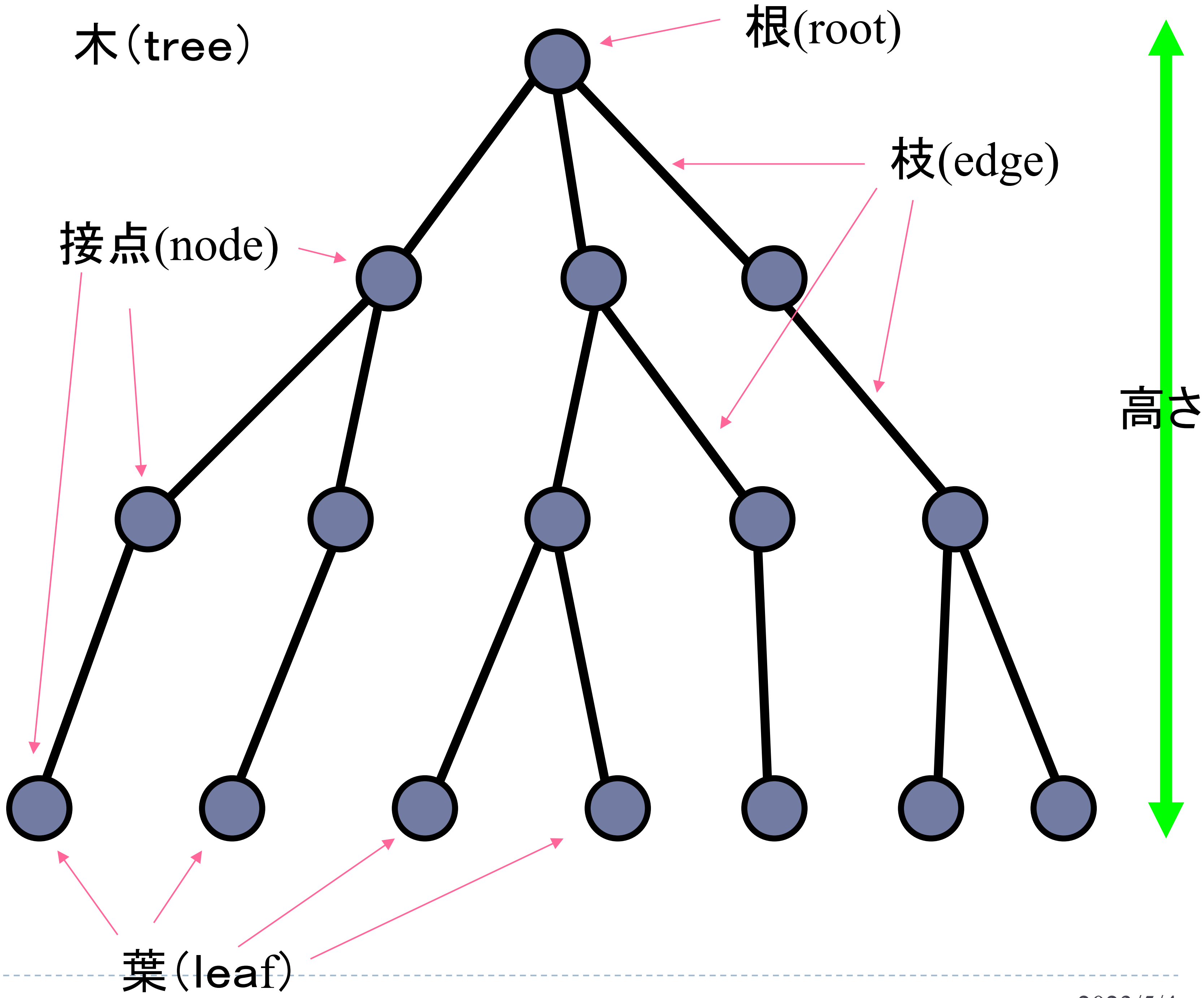


言語の等価関係の定義

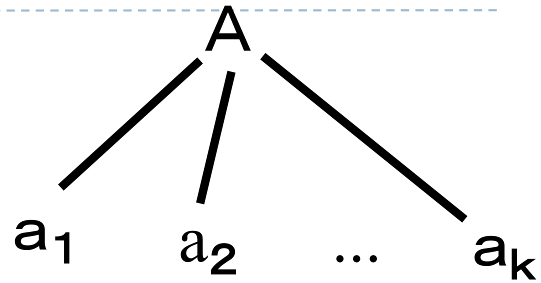
定義６:

$$L(G) = L(G')$$
$$iff$$

文法GとG’は等価


木(tree)
根(root)
枝(edge)
接点(node)
高さ
葉(leaf)

文の生成と木構造



- 文の生成

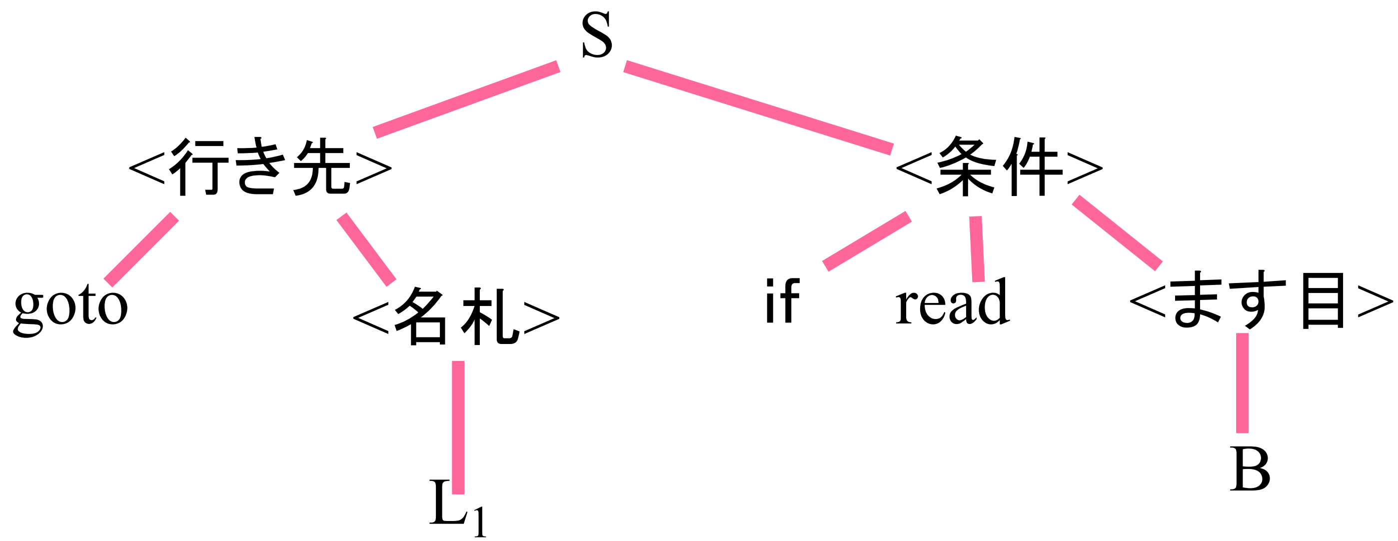
$$S = w_1 \Rightarrow w_2 \Rightarrow w_3 \Rightarrow \ldots \Rightarrow w_{n-1} \Rightarrow w_n = w$$

におけるルール$A \rightarrow a_1 a_2 \ldots a_k$の適用に対応して、
右図の構造を割り当て、全体を木構造として表現したもの。

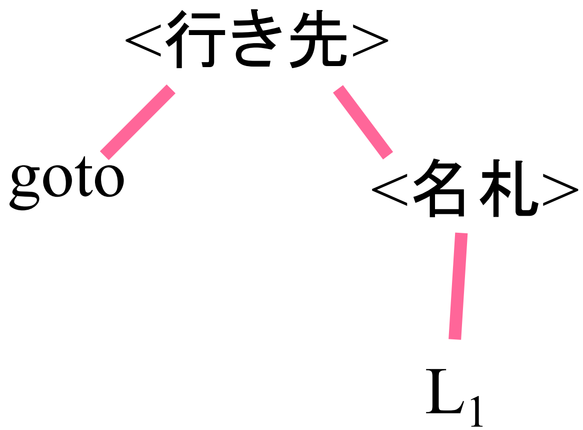
- ただ一つの根. ラベルS
- 根以外はただ一つの親を持つ. ラベル$A \in V_N$
- $A \in \Sigma$なる節は子を持たない(葉)

葉を左から読めばw
この木構造を生成木とよぶ

例) $w_1 = \text{goto } L_1 \text{ if read B}$



例) <行き先>の部分木



「あいまい」の定義(その1)

定理3：

cfg $G=(V,\Sigma,P,S)$、$S\overset{*}{\Rightarrow}w$、$w\in\Sigma^*$

iff

wのGによる生成木が存在

定義7：

cfg G_1、G_2、$L(G_1)=L(G_2)$かつ全てのwについて、G_1、G_2による生成木の構造が等しい

iff

G_1とG_2は（強）等価

「あいまい」の定義(その2)

定義8:

cfg G、任意のw∈L(G)に対し、常にただ1つの生成木

iff

Gはあいまいでない(unambiguous)

定義9:

cfl L: 本質的にあいまい

iff

Lを生成するいかなるcfg Gもあいまい

あいまいの例 1

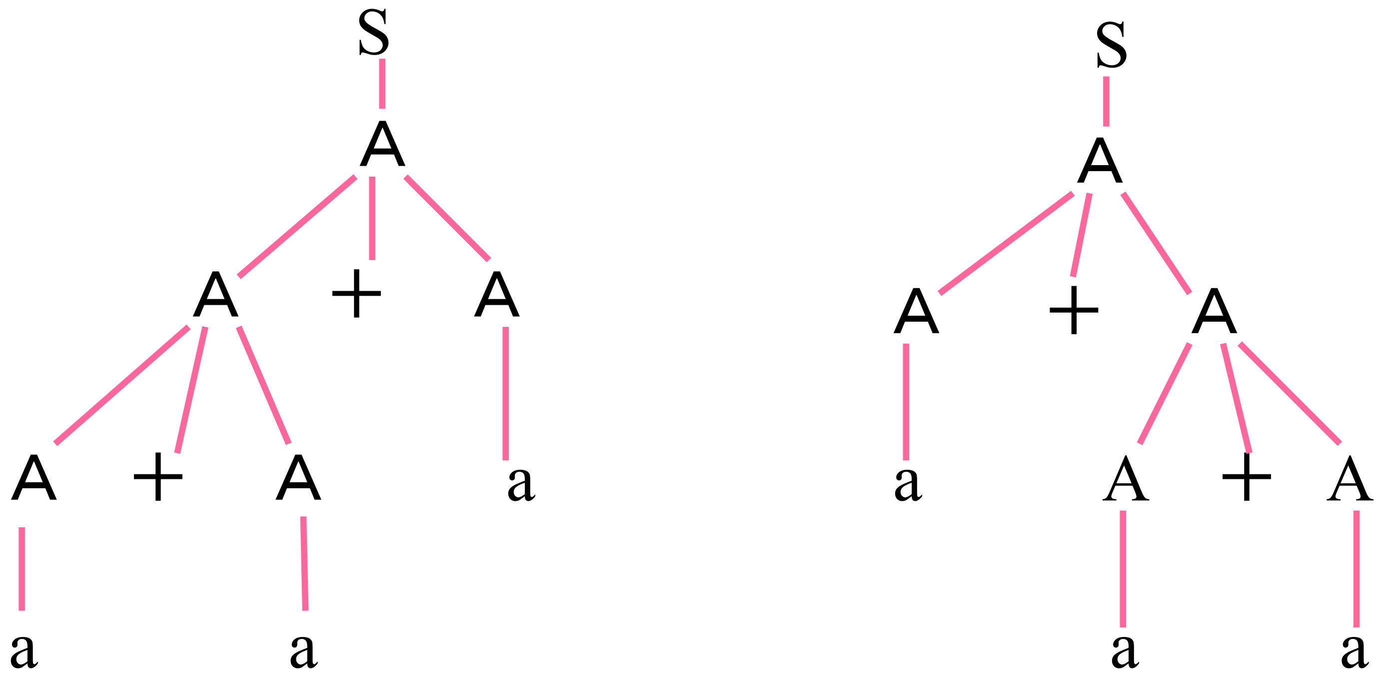
「a+a+a」に対して
二つの生成木

例） L_d=a{+a}* ← 0個以上の繰り返し

G_{d1} ＝（{S,A,a,+}, {a,+},

{S→A, A→A+A, A→a},S)は、

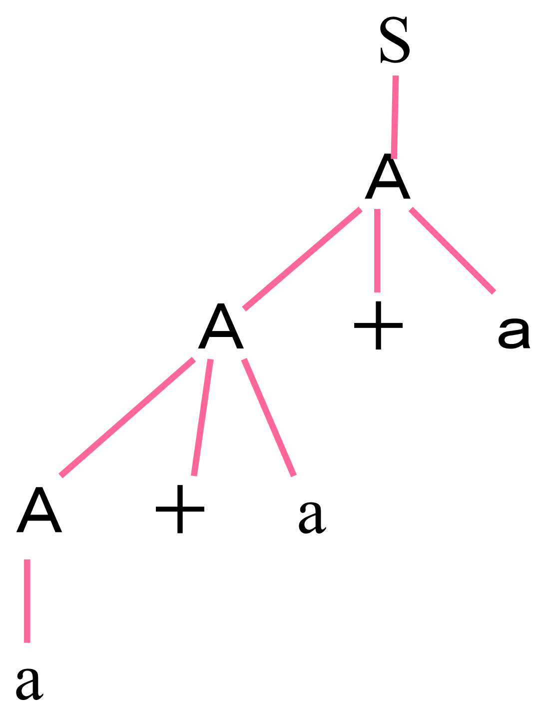
あいまい



あいまいの例２

例）G_{d2}＝({S,A,a,+}, {a,+},
{S→A, A→A+a, A→a},S)は、

あいまいでない



L_d=a{+a}*は、
本質的にあいまいでない。

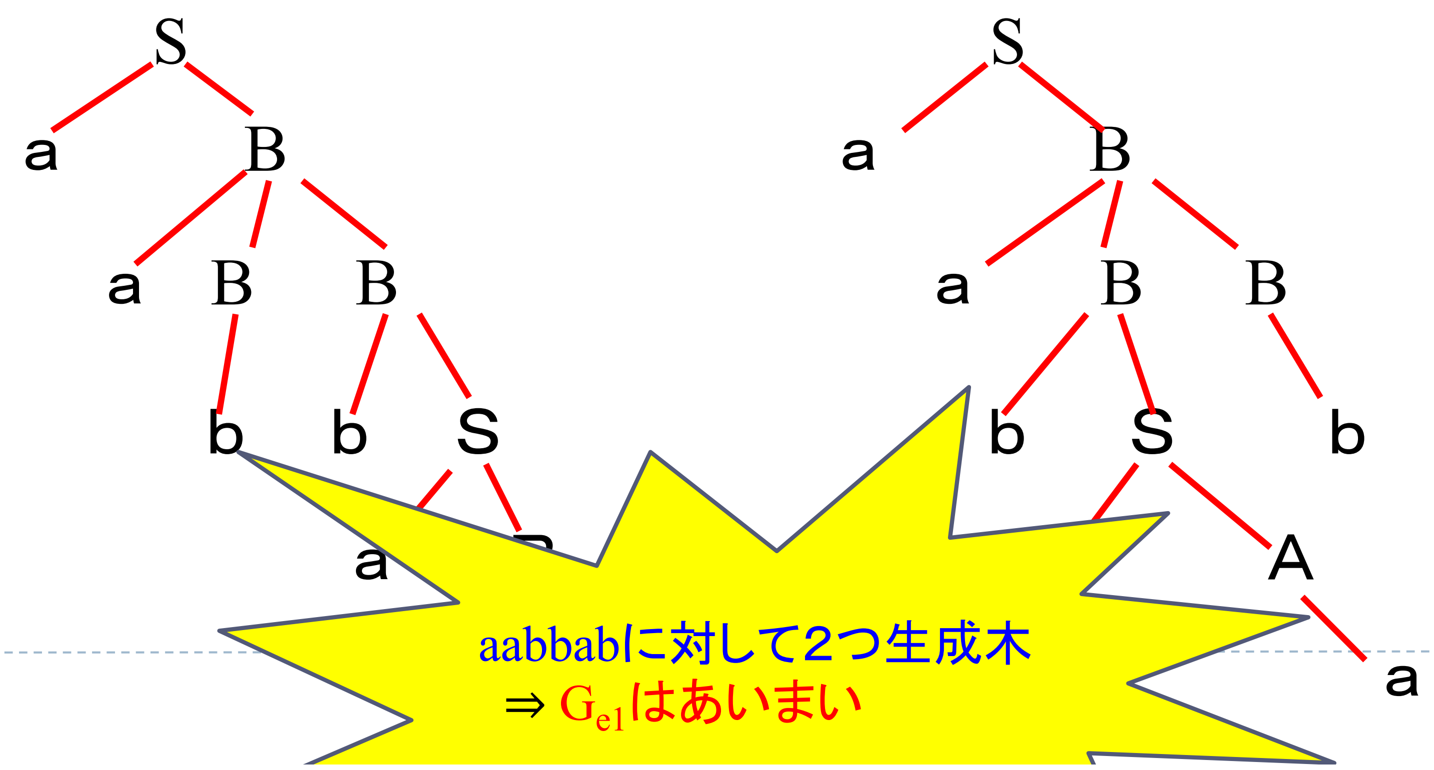
あいまいの例３

例) G_{e1}=({S,A,B,a,b},{a,b},P_{e1},S)

P_{e1} = {S→bA,A→a,A→aS,A→bAA,

S→aB,B→b,B→bS,B→aBB}

L(G_{e1})={w|wは同数のaとbを含む}

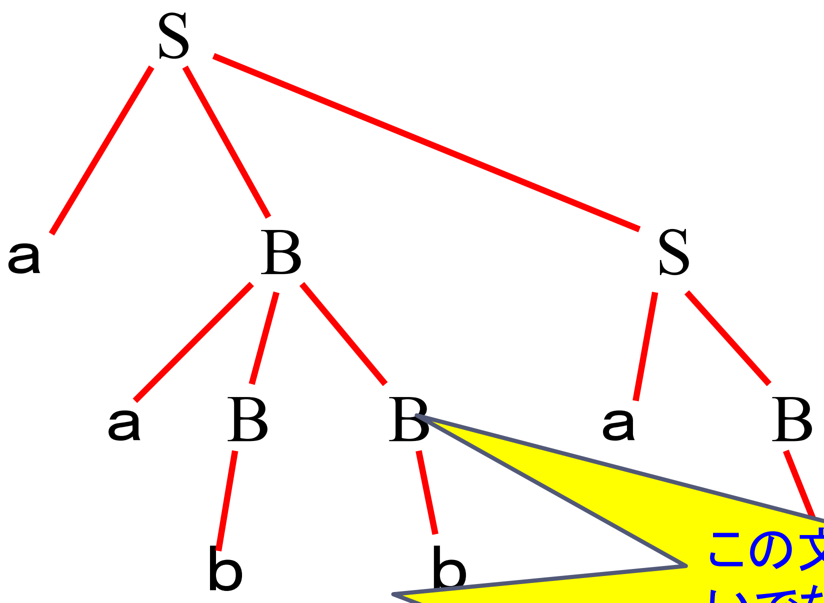


あいまいの例4

例) G_{e2}=({S,A,B,a,b},{a,b},P_{e2},S)

P_{e2} = {S→aBS,S→aB,B→aBB,B→b,

S→bAS,S→bA,A→bAA,A→a}



この文法ならすべての文であいまいでない

⇒ L={w|wは同数のaとbを含む}は本質的にあいまいでない

あいまいの例５

例)

cfl $L_f = \{a^i b^j c^k \mid i=j \vee j=k, i,j,k \geqq 1\}$

$$= \underbrace{\{a^i b^i c^k \mid i, k \geqq 1\}} \cup \underbrace{\{a^i b^k c^k \mid i, k \geqq 1\}}$$

cfgでは,別々の過程でしか生成できない

$$= \qquad L_{f1} \qquad \cup \qquad L_{f2}$$

$a^i b^i c^i \in L_{f1}$ or L_{f2}　？　　本質的にあいまい！

まとめ

1. 句構造文法の定義
   1. 定義
   2. 導出
   3. タイプ０～３
   4. 生成木
   5. あいまい性

課題

1. 文法G_bについて、次の文の導出を書け。

   aabb, aaabbb

2. 文法G_{e2}について、次の文の導出を書け。

   abab, bbaaba, ababab