計算理論　第14回
第7章：
文脈自由言語の性質 (3/3)

基礎工学部情報科学科

中川　博之

本日の概要

- 第7章：文脈自由言語の性質
  - テキスト：p.309〜
  - 7.3 文脈自由言語の閉包性
  - 7.4 文脈自由言語の決定問題

- 重要概念
  - 閉包性，決定可能性，CYKアルゴリズム

7.3 文脈自由言語の閉包性

閉包性

- 言語 L: 語の集合
- 言語クラス C: CFLなど
- L∈Cに対する演算結果がCに属するか否か

- CFLに対する演算でCFLであることが保証されているものはどのようなものか？
  – 保証されていると嬉しい場面: CFL AにCFL Bを埋め込む
  – 例）JSP(JavaServer Pages): HTML内にJavaを埋め込む
    - Javaコードを<%　%>間に記述可能
    - このとき，得られる言語がCFLだと嬉しい

7.3.1 代入

- 言語Lの各$a \in \Sigma$に対して言語L_aを対応付ける
  - Σ : 言語Lのアルファベット

- 各記号aに対する代入(substitution) : $s(a)=L_a$

- 記号列に対する代入 :
  $s(a_1 \dots a_n) = \{x_1 \dots x_n \mid x_1 \in s(a_1), \dots, x_n \in s(a_n)\}$

- 言語への代入 : $s(L) = \cup_{w \in L} s(w)$

例7.22

- $s(0)=\{a^n b^n | n \geq 1\}$, $s(1)=\{aa, bb\}$

- w=01とすると,
  $s(w)=s(0)s(1)= \{a^n b^n aa | n \geq 1\} \cup \{a^n b^{n+2} | n \geq 1\}$

- L=L(0*)のとき,
  $s(L) = s(0)^*$
  $= \{a^{n1} b^{n1} a^{n2} b^{n2} \ldots a^{nk} b^{nk} | k \geq 0 \wedge n1, \ldots, nk \geq 1\}$
  - たとえば, ε, ab, aabbabab, abaabbababなど

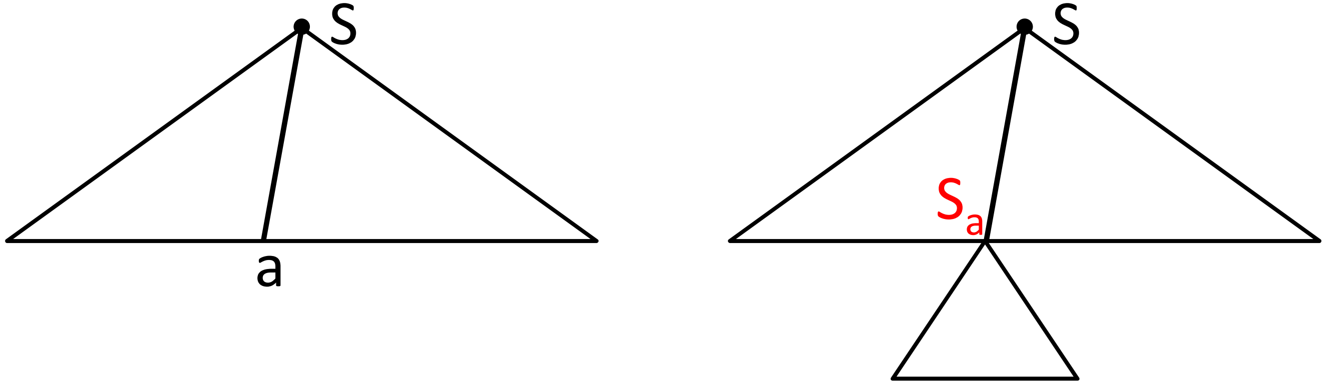
定理7.23：代入演算の閉包性

- L：Σ上のCFL
- s：Σ上の代入
- 各a∈Σに対してs(a)がCFLであれば，s(L)はCFL

定理7.23：証明(概要)

- LのCFG G=(V, Σ, P, S)
- s(a) (a∈Σ) のCFG G_a=(V_a, $Σ_a$, P_a, S_a)
  - GとG$_a$は同じ変数を使わない（置換する）
- P中の　A→... a ...　　を　A→... S_a ...　に変更

→ aではなく，Laの語が導出されるようになる



定理7.24

- 文脈自由言語は次の演算のもとで閉じている
  - (1) 集合和
  - (2) 連接
  - (3) 閉包 (*), 正閉包 (+)
  - (4) 準同型写像
- 証明(次スライド以降)
  - 代入演算の閉包性を使うとよい

定理7.24：証明（集合和）

- L_1, L_2 : CFL
- s : Σ上の代入で，$s(1) = L_1$, $s(2) = L_2$とすると

- 集合和 $L_1 \cup L_2 = s(1) \cup s(2) = s(\{1, 2\})$
  – 言語 {1, 2} への代入と定義できる

- よって，定理7.23
  – 各a∈Σに対してs(a)がCFLであれば，s(L)はCFL

より，CFLは集合和演算のもとで閉じている

定理7.24：証明（連接）

- L_1, L_2：CFL
- s：Σ上の代入で，$s(1) = L_1$, $s(2) = L_2$とすると

- 連接 $L_1 L_2 = s(1)\, s(2) = s(\{12\})$
  - 言語 {12} への代入と定義できる

- 定理7.23
  - 各a∈Σに対してs(a)がCFLであれば，s(L)はCFL

より，CFLは連接演算のもとで閉じている

定理7.24：証明（閉包，正閉包）

- L_1 : CFL
- s : Σ上の代入で，$s(1) = L_1$とすると

- 閉包 $L_1^* = s(1)^* = s(\{1^*\})$
- 正閉包 $L_1^+ = s(1)^+ = s(\{1^+\})$
  - 言語 $\{1^*\}$ および $\{1^+\}$ への代入と定義できる

- 定理7.23
  - 各a∈Σに対してs(a)がCFLであれば，s(L)はCFL

より，CFLは閉包，正閉包演算のもとで閉じている

定理7.24:証明(準同型写像)

- 準同型写像(homomorphism) (テキストp154)
  - 文字列中の各文字を特定の文字列で置き換えるような写像(関数)
    - 特定の文字列の代入

- L:CFL
- h:Σ上の準同型写像
- s:Σ上の代入で，各a∈Σに対し，s(a) = {h(a)}
とすると，

  準同型写像 $h(L)= \bigcup_{w \in L} h(w) = \bigcup_{w \in L} s(w) = s(L)$

7.3.3 : 逆順

- 定理7.25 : LがCFLなら，L^RもCFL

- 証明（一部略）
  - LのCFG G=(V, T, P, S)から G^R=(V, T, P^R, S)を作る
  - ただし，P^RはPの生成規則を逆順にしたもの
    - $A \rightarrow \alpha$ から $A \rightarrow \alpha^R$ を作る
    - 例えば，A→BC から A→CB を作る
  - $L(G^R)=L^R$は，導出の長さに関する帰納法で示す(略)

7.3.4：CFLの共通部分

- CFLは共通部分について閉じていない
  - L_1，L_2がCFLであっても，$L_1 \cap L_2$はCFLとは限らない

- 例7.26
  - $L_1 = \{0^n 1^n 2^i \mid n \geq 1, i \geq 1\}$　CFL
  - $L_2 = \{0^i 1^n 2^n \mid n \geq 1, i \geq 1\}$　CFL
  - $L_1 \cap L_2 = \{0^n 1^n 2^n \mid n \geq 1\}$　CFLではない（反復補題で証明済み）

7.3.4: 正則言語との共通部分

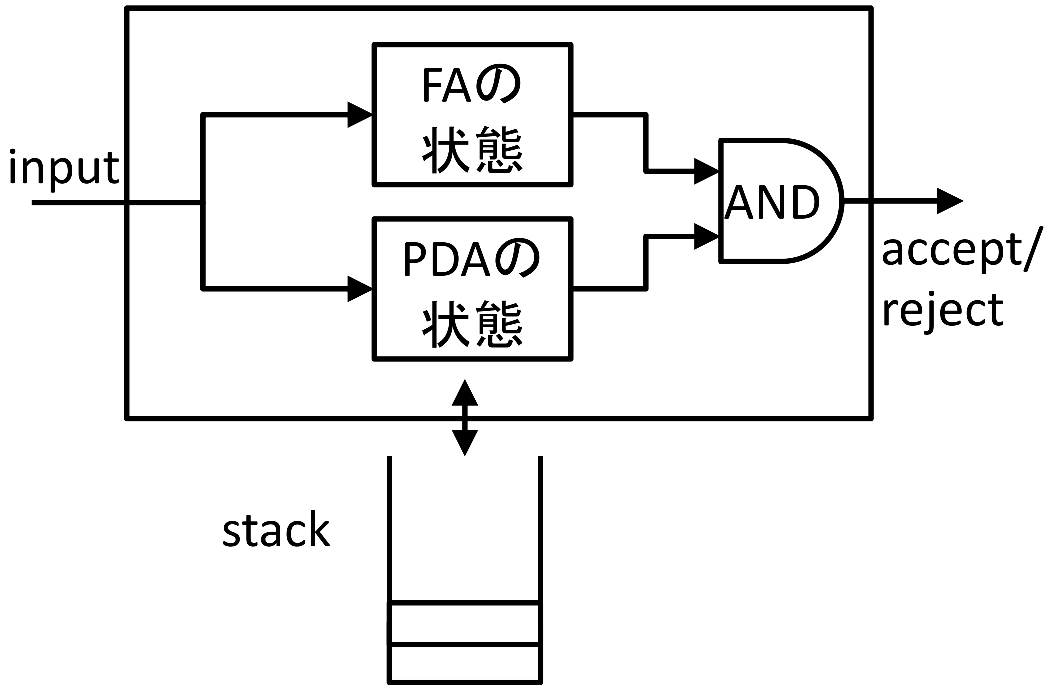
- 定理7.27:
  LがCFLでRが正則言語ならば, $L \cap R$はCFL

- 証明概要:
  – PDAとFAを用意する

PDAの構成

- 右図のようなPDAを作る
  - 上側：Rを受理するFA
  - 下側：Lを受理するPDA
  - ともに受理ならば入力を受理
  - FAとPDAの状態遷移を合成して1つのPDAを作る
  - これにより，PDAを用いて L∩R を受理できる

→ つまり，L∩RはCFL



定理7.29 (1) : CFLと正則言語の差

- L: CFL, R: 正則言語
- 定理7.29(1): L – R はCFL

- 証明:
  - Rが正則言語なら$\overline{\mathrm{R}}$も正則言語
    （テキストP147 定理4.5）
  - L – R = L∩$\overline{\mathrm{R}}$であり，CFLと正則言語の共通部分はCFL（定理7.27）である.

定理7.29 (2)：CFLの補集合

- L: CFL
- 定理7.29(2)：$\overline{L}$はCFLとは限らない

- 証明：
  - $\overline{L}$が常にCFL（補集合も閉じている）と仮定すると，
  - $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ であるが，
    - 左辺：CFLとは限らない
    - 右辺：補集合と和集合が閉じているためCFL

  となり，矛盾

定理7.29 (3) : CFLの差

- L_1, L_2: CFL
- 定理7.29(3) : L_1 - L_2はCFLとは限らない

- 証明 :
  - L_1 - L_2がCFLと仮定
  - L_1=Σ* とすると， L_1 - L_2 =$\overline{L}_2$
    - Σ*はCFL (Σ*を受理するPDA，文法を構成できる)
  - $\overline{L}_2$はCFLとは限らない一方で，仮定より L_1 - L_2がCFLとなり，矛盾

7.3.5 逆準同型写像

- L: CFL, h: 準同型写像
- $h^{-1}(L)$: $h(w) \in L$であるような列wの集合

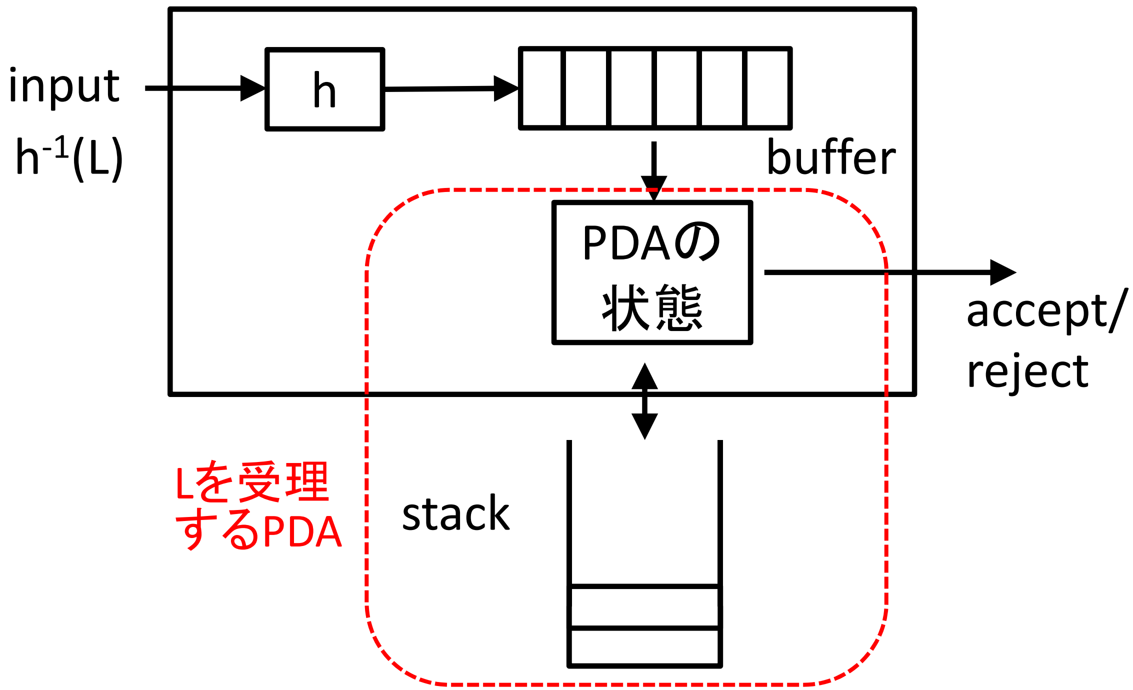
- 定理7.30 : $h^{-1}(L)$はCFL

- 証明概要
  - LはCFLなので，$h(h^{-1}(L))$を受理するPDAが存在
    - 記号列$h(a_1)h(a_2)\ldots h(a_n) \in L$を受理できる

  →内部でhを適用することで，記号列$a_1a_2\ldots a_n$を受理するPDAを構成できる
    - スタック処理を待ってもらうためにバッファを導入

7.3.5 逆準同型写像：PDAの構成

- 記号列$a_1a_2...a_n$を受理するPDAの構成
  - Lを受理するような内部PDAを用意
    - バッファから1文字ずつ読んで処理を進める
    - バッファが空なら記号a_iを読み，$h(a_i)$をバッファに挿入



7.4 文脈自由言語の決定問題

7.4.1 CFGとPDAの間の変換の複雑さ

- PDAからCFGへの変換(6.3.2参照)を考える
  - n：PDA or CFGの表現の長さ (文字列長，変数数) とすると...

- 定理7.31：長さnの記述のPDA Pから，高々$O(n^3)$の長さのCFG Gを生成する$O(n^3)$時間のアルゴリズムが存在

- 説明略(テキスト参照)

7.4.2 CNFへの変換

- (1) ε-規則の除去：
  - 素朴に実施すると$O(n^2)$，分解して上手く実施すると$O(n)$
- (2) 単位規則の除去：$O(n^2)$
- (3) 文法の到達可能性および生成的な記号の発見．つまり無用な記号の除去：$O(n)$　（後述）
- (4) 終端記号の変数への置き換え：$O(n)$
- (5) 長さ3以上の本体の分解：$O(n)$

→ 定理7.32 長さnのCFG Gに対して，Gと等価なCNF文法を$O(n^2)$で見つけることができ，結果の文法は$O(n^2)$

7.4.3 CFLの空集合検査

CFLの空集合検査問題

- G: CFL Lの文法
- S:Gの開始記号
- 判定問題:Sは列を少なくとも1つ生成するか否か？
  - つまり, Sが生成的か否かを判断
  - Sが生成的でない ⇔ L=ϕ

CFLの空集合検査アルゴリズム

- 生成的かどうかの判定法は，7.1.2(テキスト p.286)で既に学んだ
  - 終端記号からはじめて，到達可能な変数を追加していく
  - 最後に開始記号が集合に含まれているかをチェック
  - 素朴に実行すると$O(n^2)$時間
    - 各ステップですべての規則を調べるのに$O(n)$. これがnステップ
- データ構造の工夫で，$O(n)$時間で検査可能に
  - 各規則について，本体部の未確定の変数数をカウント
  - 0になると頭部の変数が生成的であると判断する手段（詳細はテキスト参照）
  - 同様の技法は，到達可能性の判定でも利用可能

7.4.4 CFLへの所属検査

CFLへの所属検査問題

- 入力: CFL L と 記号列 w
  - CFL L は PDAかCFGで与えられる
- [問題] 与えられた記号列wに対し,
  w∈Lかどうかを判別

CFLへの所属検査問題を解く CYKアルゴリズム

- CFL所属検査のための効率的なアルゴリズム
  - J. Cocke, D. Younger, T. Kasami (嵩忠雄先生: 元阪大教授)がそれぞれ発見したアルゴリズム
- $O(n^3)$時間で判定可能
  - n = |w|, PDA Pの大きさは定数とみなす
- CNF Gを用いる
  - PDAからCFGを$O(n^3)$で構成，それをCNFに$O(n^2)$で変換

CYKアルゴリズム

- 入力 $w=a_1 a_2 \ldots a_n$ （各a_i：終端記号）
- 表を作り判定
  - 各a_i：wの具体的な各文字（数はn=5のとき）
  - X_{ij}は$A \overset{*}{\Rightarrow} a_i a_{i+1} \ldots a_j$となる変数Aの集合

X_{15}				
X_{14}	X_{25}			
X_{13}	X_{24}	X_{35}		
X_{12}	X_{23}	X_{34}	X_{45}	
X_{11}	X_{22}	X_{33}	X_{44}	X_{45}
a_1	a_2	a_3	a_4	a_5

CYKアルゴリズム

- 調べたいこと：$w \in L \Leftrightarrow S \in X_{1n}$
  - 定義より，$S \in X_{1n}$は$S \overset{*}{\Rightarrow} a_1 a_2 \ldots a_n$
- 表は行ごとに下から上に埋めていく
  - 一番下は長さ1の部分列

X_{15}				
X_{14}	X_{25}			
X_{13}	X_{24}	X_{35}		
X_{12}	X_{23}	X_{34}	X_{45}	
X_{11}	X_{22}	X_{33}	X_{44}	X_{55}
a_1	a_2	a_3	a_4	a_5

表の構築法

- 基礎： 表の一番下の行 X_{ii}の構築
  - $A \overset{*}{\Rightarrow} a_i$ となる変数Aの集合
    - $A \rightarrow a_i$となるAを探せばよい
    - CNF（チョムスキー標準形）を仮定しているため

X_{15}				
X_{14}	X_{25}			
X_{13}	X_{24}	X_{35}		
X_{12}	X_{23}	X_{34}	X_{45}	
X_{11}	X_{22}	X_{33}	X_{44}	X_{55}
a_1	a_2	a_3	a_4	a_5

表の構築法

- 帰納: X_{ij} $(i<j)$ の構築
  - それより下の行はすべて計算済みと仮定
  - $A \overset{*}{\Rightarrow} a_i a_{i+1} \ldots a_j$ となる変数Aすべてを求めX_{ij}とする
    - 導出 $A \overset{*}{\Rightarrow} a_i a_{i+1} \ldots a_j$ は$A \rightarrow BC$の形で始まる
    - あるkに対し, $B \overset{*}{\Rightarrow} a_i a_{i+1} \ldots a_k$, $C \overset{*}{\Rightarrow} a_{k+1} a_{k+2} \ldots a_j$

X_{15}					
X_{14}	X_{25}				
X_{13}	X_{24}	X_{35}	←現在着目		
X_{12}	X_{23}	X_{34}	X_{45}		←計算済み
X_{11}	X_{22}	X_{33}	X_{44}	X_{55}	
a_1	a_2	a_3	a_4	a_5	

表の構築法

- 帰納: X_{ij} $(i<j)$ の構築
- X_{ij}は以下の条件を満たす変数Aの集合
  - A→BCがGの規則
  - $i \le k < j$を満たすi, j, kそれぞれに対して
    - $B \in X_{ik}$　: $B \stackrel{*}{\Rightarrow} a_i a_{i+1} \ldots a_k$
    - $C \in X_{k+1\,j}$: $C \stackrel{*}{\Rightarrow} a_{k+1} a_{k+2} \ldots a_j$
- つまり，Aを求めるには以下を調べればよい
  - $(X_{ii}, X_{i+1\,j}), (X_{ii+1}, X_{i+2\,j}), \ldots, (X_{i\,j-1}, X_{jj})$
  - これらのXはいずれも計算済

例7.34

- G:CNF
  - S → AB | BC
  - A →BA | a
  - B →CC | b
  - C →AB |a
- baaba ∈ L(G)?

X_{15} {S,A,C}				
X_{14} -	X_{25} {S,A,C}			
X_{13} -	X_{24} {B}	X_{35} {B}		
X_{12} {S, A}	X_{23} {B}	X_{34} {S, C}	X_{45} {S, A}	
X_{11} {B}	X_{22} {A, C}	X_{33} {A, C}	X_{44} {B}	X_{55} {A, C}
b	a	a	b	a

7.4.5 決定不能なCFL問題のあらまし

決定可能性 (decidability)

- 決定可能な(decidable)問題
  - 有限時間で答(Yes/No)を出力して停止するアルゴリズムが存在する問題
- 決定不能な(undecidable)問題
  - 決定可能でない問題. すなわち,
  - 判定に要する時間が有限時間でない問題

CFLに関する決定不能な問題

- 与えられたCFGはあいまいか
- 与えられたCFLは本質的にあいまいか
- 2つのCFLの共通部分は空集合か
- 2つのCFLが等しいか
- 与えられたCFLがΣ^*に等しいか

ミニレポート

ミニレポート：14-1

- テキストp331 問7.4.3 (a)：
  - 例7.34の文法GとCYKアルゴリズムを使い，ababaがL(G)に属するか決定せよ

- 文法G=({S, A, B, C}, {a, b}, P, S)

  ただし Pは以下の生成規則を要素とする集合
  - S→AB|BC
  - A→BA|a
  - B→CC|b
  - C→AB|a