計算理論　第11回
第6章：
プッシュダウンオートマトン (2/2)

基礎工学部情報科学科
中川　博之

本日の概要

- 第6章：プッシュダウンオートマトン
  - テキスト：p.265～
  - 6.3 PDAとCFGの等価性
  - 6.4 決定性PDA

- 重要概念
  - CFGとの等価性，決定性PDA

6.3 PDAとCFGの等価性

等価性の証明方針

- CFGが受理するクラス = PDAが受理するクラス

- 以下の方針で示す

  １．任意のCFG Gに対し，Gの言語を受理するPDAが存在する

  - CFGのクラス ⊆ PDAが受理する言語のクラス

  ２．任意のPDA Pに対し，Pが受理する言語はCFG

  - PDAが受理する言語のクラス ⊆ CFGのクラス

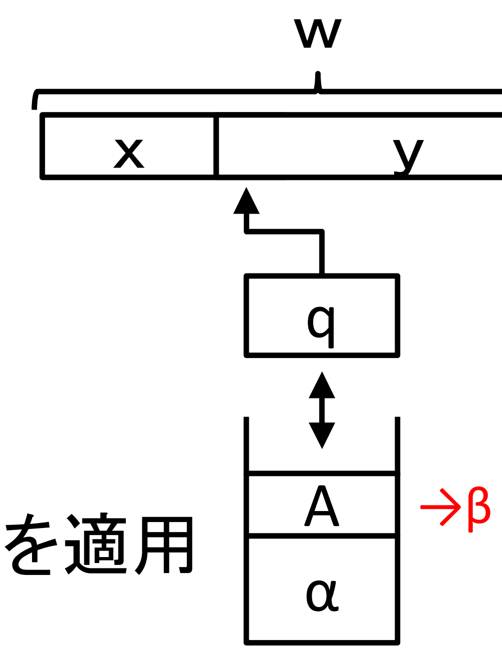
文法からPDAへ

- 入力: 任意のCFG G
- 出力: Gの言語を空スタック受理するPDA P

- 変換アルゴリズムの方針
  - PDA PはGの最左導出を模倣
  - 模倣の上，L(G)に含まれる文字列wが導出できればPは受理

文法Gの言語を受理するPDAの概要 (1/2)

- Gでの導出の途中 : $S \overset{*}{\Rightarrow} xA\alpha$
  - A : 最も左に現れる変数
  - xは変換が終わった終端記号だけの列（最左導出なので）
  - α : 変数/終端記号を含む列
- 対応するPDAの動作 :
  - 入力記号列w
  - x : 読み終えた終端記号列
  - y:残りの文字列 (w=xy)
  - スタック上端Aに対応する規則A→βを適用
    - $(q, y, A\alpha) \vdash (q, y, \beta\alpha)$



文法Gの言語を受理するPDAの概要 (2/2)

- もしスタック上端が終端記号なら，yの先頭文字と照合して，一文字読み進める
  - スタック上端の終端記号を消費

- 全体概要
  - 動作開始時：Gの出発記号Sをスタックにプッシュ
  - 入力記号列wを読み終えたときに空スタックなら受理

文法Gの言語を受理するPDA

- CFG $G=(V, T, Q, S)$としたとき，言語$L(G)$を<u>空スタック受理</u>するPDA Pは，

$$P=(\{q\}, T, V\cup T, \delta, q, S)$$

- ただし，
  - 各変数Aについて　　　　　　　　　↓生成規則を適用
    $\delta(q, \varepsilon, A) = \{(q, \beta) \mid A \rightarrow \beta$がQの生成規則$\}$
  - 各終端記号aについて
    $\delta(q, a, a) = \{(q, \varepsilon)\}$　　　←照合して消費

※テキスト p266 (1) 内 誤: A→βがPの生成規則 → 正： A→βがQの生成規則

例6.12

- CFG G_{Exp}=(V, T, Q, S)
  - V={I, E}, T={a,b,0,1,(,),+,*}, S=E
  - I → a | b | Ia | Ib | I0 | I1
  - E→ I | E*E |E+E | (E)
- PDA P=({q}, T, V∪T, δ, q, S)
  - δ(q, ε, I)={(q, a), (q, b), (q, Ia), (q, Ib), (q, I0), (q, I1)}
  - δ(q, ε, E)={(q, I), (q, E+E), (q, E*E), (q, (E))}
  - δ(q, a, a)={(q, ε)}, δ(q, b, b)={(q, ε)}, ...

※テキスト p266下から2行目b）誤: (q,ε,E) → 正 : δ(q,ε,E)

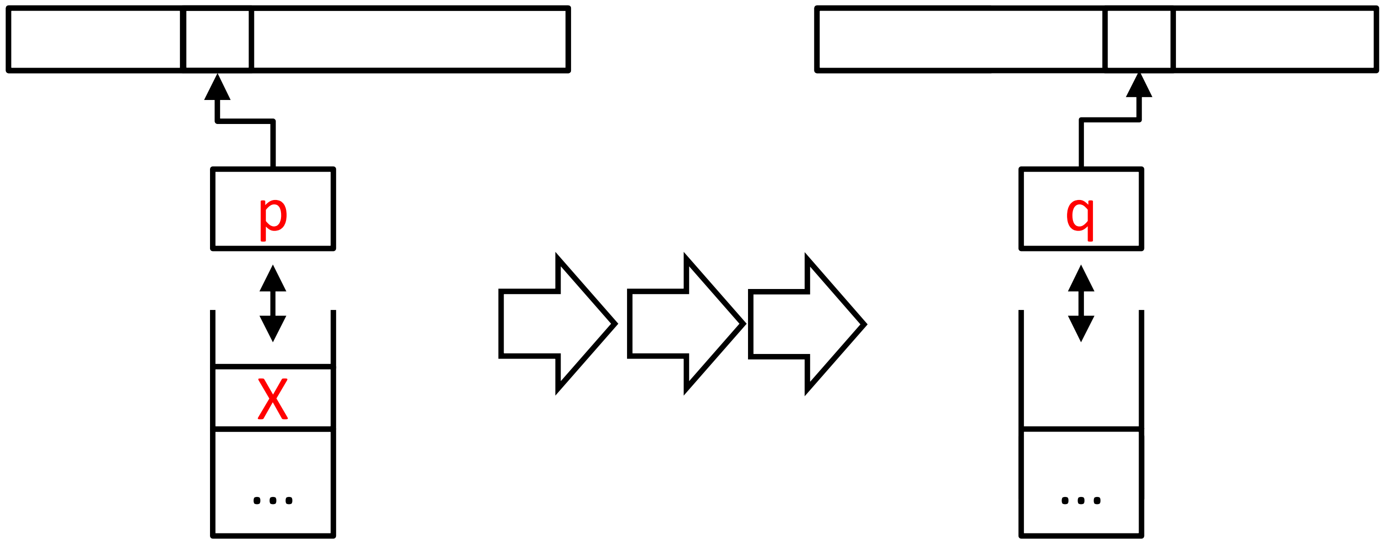
PDAから文法へ

- 入力: 任意の<u>空スタック受理</u>PDA P
- 出力: Pが受理する言語を生成するCFG G

- 変換アルゴリズムの方針
  - Pの遷移関数の各要素に対応した生成規則を作成

文法上の変数

- 文法上の各変数は [pXq] の形
  - これで1つの変数
- [pXq]はPDA Pの以下の動作に対応
  - スタック上端の記号Xが消費される
  - Xを消費するまでに状態がpからqに遷移



PDAからの文法構成(1/2)

- S:開始記号
  - これだけ3つ組ではない
- Sに対する生成規則 $S \rightarrow [q_0\ Z_0\ p]$ を追加
  - q_0 : PDAの初期状態
  - Z_0 : PDAの初期スタック記号
  - p : 任意の状態(すべての状態に対して用意する)
    - 空スタック受理でどの状態で受理しても良いため
- 変数 $[q_0\ Z_0\ p]$ は$(q_0, w, Z_0) \vdash^* (p, \varepsilon, \varepsilon)$を表現
  - 状態q_0から何回かの遷移を経てスタック上端のZ_0を消費して，状態pに至り，空スタック受理

PDAからの文法構成(2/2)

- PDAが遷移関数$(r, Y_1Y_2\ldots Y_k) \in \delta(q, a, X)$を持つ場合，以下の生成規則を文法Gに追加

$$[qXr_k] \rightarrow a[rY_1r_1][r_1Y_2r_2]\ldots[r_{k-1}Y_kr_k]$$

- PDAの動作としての解釈：
  - 状態qから入力aを読んで，スタック上端のXを$Y_1Y_2\ldots Y_k$に書き換えて状態rへ (a=εの場合もあり)
  - その後，追加した$Y_1Y_2\ldots Y_k$を消去する必要がある
    - 最終的に全て消去できたときの状態がr_k

PDAと文法との関係

- [性質] $[qXp] \overset{*}{\Rightarrow} w$であるための必要十分条件は $(q, w, X) \vdash^* (p, \varepsilon, \varepsilon)$
  - 証明はテキストp271～272参照

これを使うと…

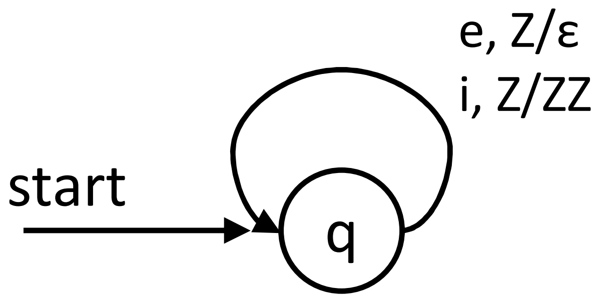
- 先に生成規則 $S \rightarrow [q_0 Z_0 p]$ を追加したが，
- $S \overset{*}{\Rightarrow} w$であるとき，そのときのみに限り

$$(q_0, w, Z_0) \vdash^* (p, \varepsilon, \varepsilon)$$

- つまり，空スタック受理できるときのみ，$S \overset{*}{\Rightarrow} w$

例6.15

- PDA $P_N=(\{q\}, \{i,e\}, \{Z\}, \delta_N, q, Z)$
  - $\delta_N(q, i, Z)=\{(q, ZZ)\}$, $\delta_N(q, e, Z)=\{(q, \varepsilon)\}$
  - if/elseの誤りを空スタックで受理するPDA
    - elseがifを上回ると受理（エラー時に受理（検知））



例6.15：文法Gの生成

- 変数は2つ：開始記号 S と [qZq]
  - 状態もスタック記号も1つずつしかないため
- 生成規則

  $\delta(q,a,X) \ni (r, Y_1Y_2 \ldots Y_k)$のとき

  $[qXr_k] \rightarrow a[rY_1r_1][r_1Y_2r_2] \ldots [r_{k-1}Y_kr_k]$

  - $S \rightarrow [qZq]$
  - $\delta_N(q, i, Z)=\{(q, ZZ)\}$より　$[qZq] \rightarrow i[qZq][qZq]$
  - $\delta_N(q, e, Z)=\{(q, \varepsilon)\}$より　$[qZq] \rightarrow e$
- [qZq]をAで表すと
  - $S \rightarrow A$
  - $A \rightarrow iAA \mid e$
  - SとAは同一視できるので，$G=(\{S\}, \{i,e\}, \{S \rightarrow iSS \mid e\}, S)$

6.4 決定性PDA

決定性PDA (DPDA)

- 決定性プッシュダウンオートマトン
  (Deterministic pushdown automaton : DPDA)

  – 受理できる言語は文脈自由言語の部分クラス
  – コンパイラの構文解析器はDPDA

DPDAの定義

DPDAは次の2つの条件によって定義する

- $\forall q \in Q$, $\forall a \in \Sigma \cup \{\varepsilon\}$, $\forall X \in \Gamma$に対して
  <u>$\delta(q,a,X)$は高々一つの要素を含む</u>
  – 1文字読み動作は高々1通りだけ

- Σ中のあるaに対して,
  <u>$\delta(q,a,X)$が空でなければ, $\delta(q,\varepsilon,X)$は空</u>
  – 1文字読み動作かε動作のいずれかのみ

例6.16

- 言語L_{wwr} = {ww^R|wは(0+1)*に属する}はCFLだがDPDAでは受理できない
  - 以前述べた通り（非決定的）PDAだと受理できる（境界を非決定的に推測する）が，DPDAでは境目を判断することが出来ない
- 言語L_{wcwr} = {wcw^R|wは(0+1)*に属する}ならDPDAでも受理できる
  - w中の文字をスタックに順次積む
  - cを見つけたら照合用の状態に遷移
  - w^Rの文字とスタック上端の文字を順次照合

正則言語とDPDA

- 定理6.17の概要:
  - 任意の正則言語Lに対して，Lを受理するDPDA Pが存在
- 証明
  - スタックは使わない
  - DFA A=(Q, Σ, δ_A, q_0, F)のとき
  - DPDA P =(Q, Σ, {Z_0}, δ_F, q_0, Z_0, F) と構成すればよい
    - もしδ_A(q, a)=pならδ_F(q, a, Z_0)={(p, Z_0)}と定義する

空スタック受理DPDAの言語認識能力

- ただし空スタック受理DPDAは，言語認識能力がかなり限定される

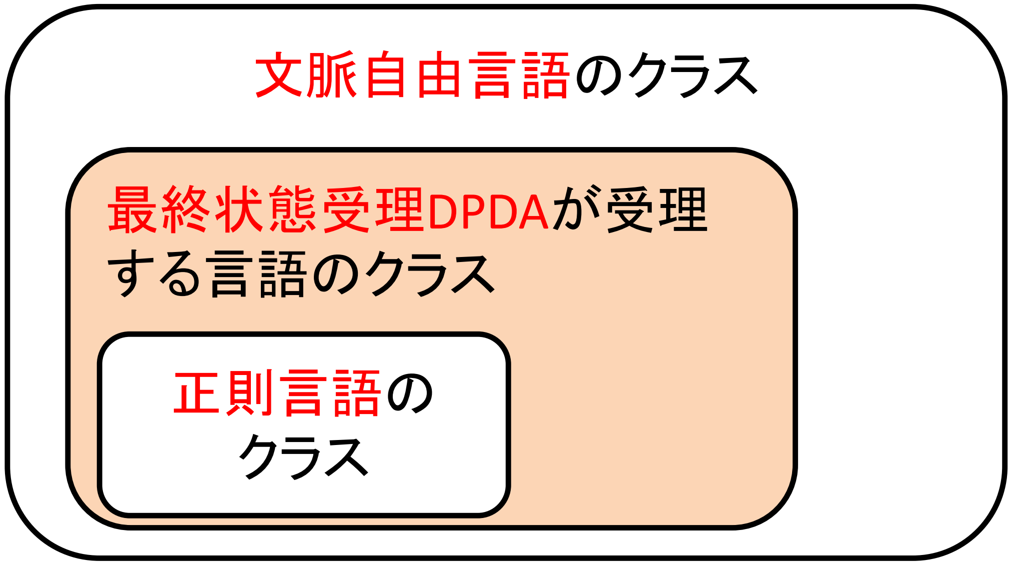
→ prefix性を持たない言語は受理できない

  - prefix性を持つ言語L：Lのどの列x,y (x≠y)も，一方が他方のprefixではない

- 例えば，{0}*は，
  - prefix性を持たない
  - 正則言語であるが
  - 空スタック受理DPDAでは受理できない
    - どこで空スタックにすべきかを判断できない

DPDAと文脈自由言語

- 今までの議論から
  - 任意の正則言語Lに対して，Lを受理するDPDA P が存在（定理6.17）
  - CFLにはDPDAでは受理できない言語もある（例6.16の言語L_{wwr}）



最終状態受理DPDAが受理する言語のクラス：
・正則言語のクラスを真に含む
・文脈自由言語のクラスに真に含まれる

DPDAとあいまいな文法

- DPDAが受理する言語は，「本質的にあいまいでない文脈自由言語」の部分クラス
  - 本質的にあいまいでないCFLのすべてを受理できるわけではない（例：L_{wwr}）
  - 定理6.20：あるDPDA Pについて，L=N(P)ならばLはあいまいでない文脈自由文法で記述できる
  - 定理6.21：あるDPDA Pについて，L=L(P)ならばLはあいまいでない文脈自由文法で記述できる

N(P)：空スタック受理，L(P):最終状態受理

ミニレポート

ミニレポート：11-1

- テキストp280 問6.4.2 a)を一部改訂
  - 次の言語を受理する決定性プッシュダウン・オートマトン(DPDA)を作れ．ただし，最終状態で受理するDPDAとせよ．
  - $\{0^n1^m \mid n \leq m\}$