アルゴリズム論 1
第 7 回
プッシュダウンオートマトン (2)

関川 浩

2017/05/31

第 4 回から第 7 回の目標

正規表現と fa: よくできたシステムだが能力が低い

より能力が高いシステムを導入する

- 文脈自由文法 (第 4, 5 回)
- プッシュダウンオートマトン (第 6, 7 回)

第 7 回の目標:

- プッシュダウンオートマトンの設計 (前回からの続き)
- 文脈自由文法とプッシュダウンオートマトンの等価性

1. プッシュダウンオートマトンの設計 (前回からの続き)
    - 例題 3
    - pda による fa の模倣
    - 例題 4
2. 文脈自由文法とプッシュダウンオートマトンの等価性
    - cfg と pda の等価性の証明の方針
    - 例題 5
    - 状態数 1 の pda
    - cfg と pda の等価性

1. プッシュダウンオートマトンの設計 (前回からの続き)
2. 文脈自由文法とプッシュダウンオートマトンの等価性

例題 3 (1/6)

例題 3

$\{0^m 1^n \mid m, n \geq 1 \ (m \neq n)\}$ を認識する pda を構成せよ

解答 (1/5)

以下で言及しないパターンの場合は停止して非受理

たとえば:

- 入力記号列が ε の場合
- $m > n$ とゲスした場合の記号ゲスモードで 1 を読んだ場合
- $m < n$ とゲスした場合のポップモードで 0 を読んだ場合
- $\ldots$

例題 3 (2/6)

解答 (2/5)

最初に $m > n$ であるか $m < n$ であるかをゲス

($m = n$ の場合はどちらもはずれなので非受理)

(1) $m > n$ (0 の方が多い) とゲスした場合

スタックに触らず入力ヘッドを動かして (これで $m = n$ を排除)
記号ゲスモードに入る

- 記号ゲスモード
  現在読んでいる 0 が右端の 0 から n 番目か否かをゲス

  Yes: スタックの Z_0 を 0 に書き換え 0 チェックモードへ移行
  No: 記号ゲスモードを続行

$$0 \ldots\ldots 0\ \overbrace{0 \ldots\ldots 0}^{n\,個}\ \overbrace{1 \ldots\ldots 1}^{n\,個}$$

右端の 0 から n 番目とゲス

例題 3 (3/6)

解答 (3/5)

- 0 チェックモード

  0 を読んだとき: 0 をスタックにプッシュし現モードを続行

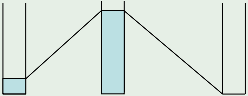
  1 を読んだとき: ポップして 1 チェックモードへ移行

- 1 チェックモード

  1 を読んだとき: ポップして現モードを続行

$$\boxed{0 \ldots\ldots 0\ \overbrace{0 \ldots\ldots 0}^{n\,\text{個}}\ \overbrace{1 \ldots\ldots 1}^{n\,\text{個}}}$$

スタック動作後ヘッド移動前のスタックの状態



例題 3 (4/6)

解答 (4/5)

(2) $m < n$ (1 の方が多い) とゲスした場合

スタックの Z_0 を 0 に書き換え, 入力ヘッドを動かして積み上げモードへ

- 積み上げモード

  0 を読んだとき: 0 をスタックに積み上げ, 現モード続行

  1 を読んだとき: スタックに触らず入力ヘッドを動かして (これで $m = n$ を排除) 記号ゲスモードへ

例題 3 (5/6)

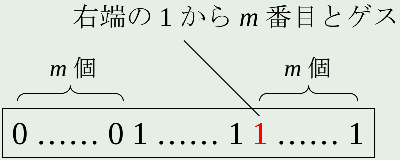
解答 (5/5)

- 記号ゲスモード

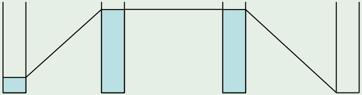
読んでいる 1 が右端の 1 から m 番目か否かをゲス

Yes: その 1 からポップモードに入って, 1 を読むたびにポップ

No: 記号ゲスモードを続行



スタック動作後ヘッド移動前
のスタックの状態



例題 3 (6/6)

注意

もし,

$\{0^m 1^n \mid m, n \geq 0 \ (m \neq n)\}$ を認識する pda を構成せよ

とすると,

- 0^m ($n = 0$ の場合)
- 1^n ($m = 0$ の場合)

も受理しなければいけないので, 複雑になる

pda による fa の模倣

pda: fa に補助記憶装置を追加したもの $\Longrightarrow$ fa を模倣できる

受理条件に注意: 入力を読み終わったときにちょうどスタックが空
しかし, 今読んでいる記号がテープの右端か否かは不明

解決策: ゲスを利用

M: 与えられた dfa
T: M を模倣する pda

T はスタックには触らず, M の状態遷移を模倣しながら, 現在の記号がテープの右端か否かをゲス

Yes: その記号を読んだ行先が M の受理状態なら Z_0 をポップ
No: 模倣を続行

注: pda が決定性なら fa を模倣できない (ゲスが使えない)
$\Longrightarrow$ pda は決定性と非決定性で言語を認識する能力が異なる

例題 4 (1/7)

例題 4

$L = \{x \mid x \in \{a,b\}\{a,b\}^*$ かつ $x = yy$ と書けない $\}$ を認識する pda M を構成せよ

注意

$\{a,b\}^* \setminus L = \{yy \mid y \in \{a,b\}^*\}$ は文脈自由言語ではない (第 5 回の例題 3)

解答 (1/5)

$x \in L \iff x$ は以下のいずれかの条件を満たす

(i) $|x|$ は奇数

(ii) $|x|$ は偶数で, $x = a_1 \dots a_n a_{n+1} \dots a_{2n}$ としたとき, ある i に対して $a_i \neq a_{n+i}$

(ii) の異なる記号を d_1, d_2 とする

例題 4 (2/7)

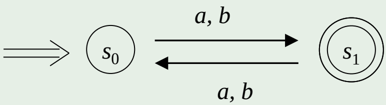
解答 (2/5)

M は最初に条件 (i), (ii) のどちらが満たされるかゲスする

(1) 条件 (i) を満たすとゲスした場合

奇数チェックモードに入る

$|x|$ が奇数か否かは fa でチェック可能なので, それを模倣



例題 4 (3/7)

解答 (3/5)

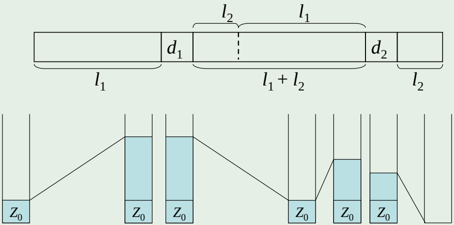
(2) 条件 (ii) を満たすとゲスした場合 (1/3)

- d_1 ゲスモード
  M は現在の記号が d_1 か否かをゲス

  No: 1 個の記号をプッシュしてこのモードを続行

  Yes: その記号が a か b かを有限状態を利用して記憶し，
  スタックには触らず d_2 ゲスモードへ

スタック動作後
ヘッド移動前の
スタックの状態



例題 4 (4/7)

解答 (4/5)

(2) 条件 (ii) を満たすとゲスした場合 (2/3)

- d_2 ゲスモード

  Z_0 が現れるまでポップ

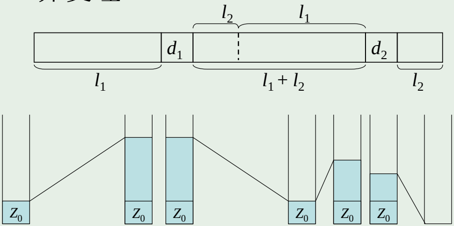
  Z_0 が現れたら d_1 のときと同様に d_2 をゲス

  No: 1 個の記号をプッシュしてこのモードを続行

  Yes: 記憶している d_1 の値と違っていればポップして排出モードへ

  同じなら停止して非受理

スタック動作後
ヘッド移動前の
スタックの状態



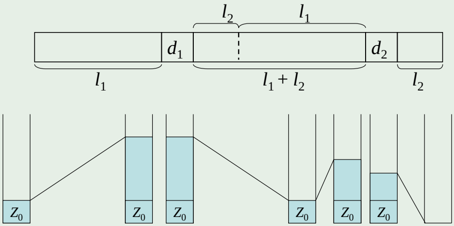
例題 4 (5/7)

解答 (5/5)

(2) 条件 (ii) を満たすとゲスした場合 (3/3)

- 排出モード
  記号を一つ読むごとにポップ

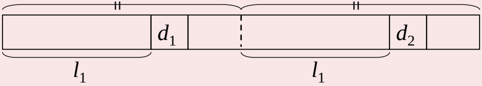
スタック動作後
ヘッド移動前の
スタックの状態



例題 4 (6/7)

注意 (1/2)

(ii) で, d_1 をゲスしたあと, 中央右隣をゲスする方法 (自然な方法) は不可



- 両方のゲスが当たったときは問題ない
- 中央右隣のゲスがはずれた場合が問題

(例) 受理してはいけない列 $abaaaabaaa$
左から 2 番目の b を d_1 とゲス
5 番目の a を中央右隣とゲス
$\Longrightarrow$ 受理 (6 番目の a が d_2)

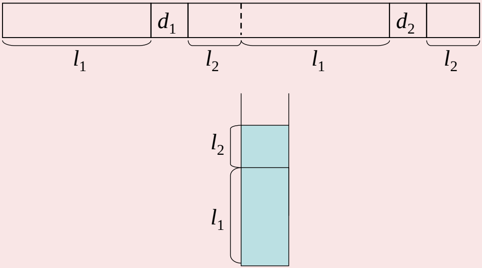
例題 4 (7/7)

注意 (2/2)

中央右隣のゲスがはずれたことが確認できれば問題ない

$\Longrightarrow$ 中央左隣までの記号数をスタックに蓄える必要がある

$\Longrightarrow$ l_1 の情報が取り出せなくなってしまう



1. プッシュダウンオートマトンの設計 (前回からの続き)

2. 文脈自由文法とプッシュダウンオートマトンの等価性

cfg と pda の等価性の証明の方針

- pda の重要性: cfg との等価性
  受理条件や非決定性も cfg に合わせるため
- cfg と pda の等価性の証明
  pda の状態数は 1 で十分であることを示す
  アイディア: スタック記号に状態の情報を載せる

  1 状態の pda は cfg とほとんど同じ

例題 5 (1/3)

例題 5

$\{xx^{\mathrm{R}} \mid x \in \{a,b\}\{a,b\}^*\}$ を認識する 1 状態の pda を構成せよ

解答 (1/2)

前回: 2 状態 s_0, s_1 で積み上げモードとチェックモードを区別

今回: スタックの先頭記号で区別

- 積み上げモード: A', B' (先頭より下は A, B)
- チェックモード: A, B

例題 5 (2/3)

解答 (2/2)

状態遷移関数 (以下にないものの値は $\emptyset$)

$$\delta(s_0, a, Z_0) = \{(s_0, A')\}, \quad \delta(s_0, b, Z_0) = \{(s_0, B')\},$$
$$\delta(s_0, a, A') = \{(s_0, A'A), (s_0, \varepsilon)\}, \quad \delta(s_0, b, A') = \{(s_0, B'A)\},$$
$$\delta(s_0, a, B') = \{(s_0, A'B)\}, \quad \delta(s_0, b, B') = \{(s_0, B'B), (s_0, \varepsilon)\},$$
$$\delta(s_0, a, A) = \{(s_0, \varepsilon)\}, \quad \delta(s_0, b, B) = \{(s_0, \varepsilon)\}$$

1 状態になると cfg とほとんど同じ

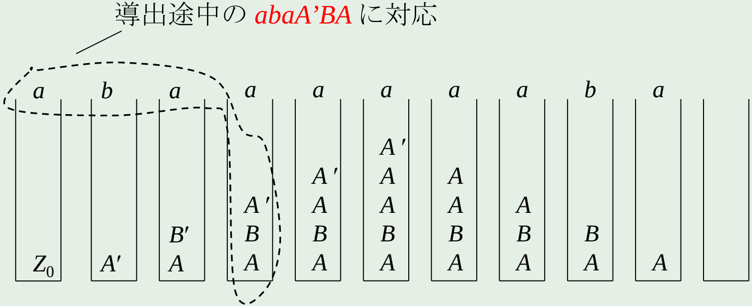
対応: $(s_0, \alpha) \in \delta(s_0, c, D) \iff D \to c\alpha$

$$Z_0 \to aA', \quad Z_0 \to bB',$$
$$A' \to aA'A,\ A' \to a, \quad A' \to bB'A,$$
$$B' \to aA'B, \quad B' \to bB'B,\ B' \to b,$$
$$A \to a, \quad B \to b$$

例題 5 (3/3)

導出例:

$$Z_0 \Rightarrow aA' \Rightarrow abB'A \Rightarrow abaA'BA \Rightarrow abaaA'ABA$$
$$\Rightarrow abaaaA'AABA \Rightarrow abaaaaAABA \Rightarrow abaaaaaABA$$
$$\Rightarrow abaaaaaaBA \Rightarrow abaaaaaabA \Rightarrow abaaaaaaba$$



定理 1

与えられた pda に対し，同じ言語を認識する 1 状態の pda が構成可能

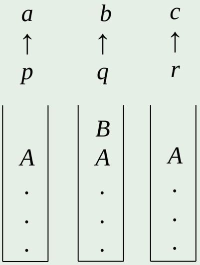
証明のアイディア (1/3)

スタック記号に情報を載せるアイディアのみ示す

M: 与えられた pda

M が入力記号 a, b, c を

- 状態を p, q, r と推移しながら読み
- その間にスタックに B をプッシュし直後にポップしたとする



定理 1 (2/3)

証明のアイディア (2/3)

M_s: M を模倣する 1 状態の pda

- スタック記号は (s_1, C, s_2)

  s_1, s_2: M の状態

  C: M のスタック記号

- M の状態が q で B をプッシュ

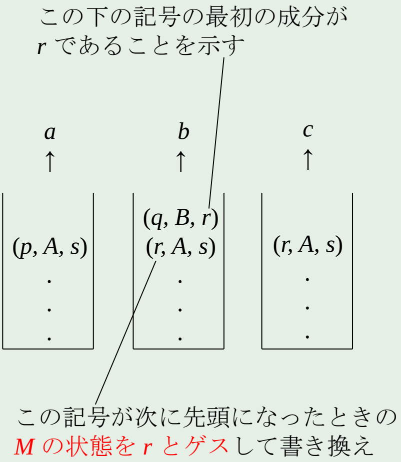
  $$\Updownarrow$$

  M_s では (q, B, r) をプッシュ

  同時に, それまでの先頭記号の

  第 1 成分を書き換え

  (この記号が次に先頭になったときの M の状態をゲス)



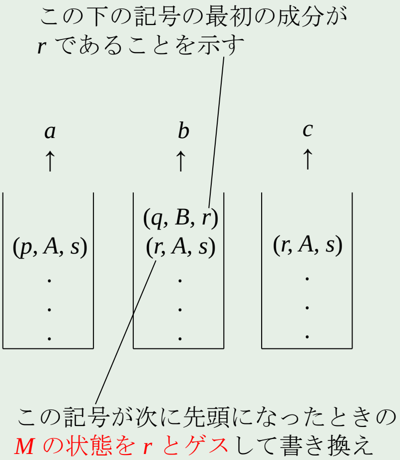
定理 1 (3/3)

証明のアイディア (3/3)

第 3 成分: ゲスの判定に必要

- 第 1, 2 成分と入力記号から次状態を求めて第 3 成分と比較
- 第 3 成分がないとゲスの判定にポップが必要

  ポップすると正解が不明に ■



cfg と pda の等価性 (1/2)

定理 2

cfg と pda は等価

証明 (1/2)

- pda M から cfg G を構成

  M を 1 状態の pda $M_1 = (\{s\}, \Sigma, \Gamma, \delta, s, Z_0)$ に直す

  $G = (\Gamma, \Sigma, P, Z_0)$, ただし

  $$P = \{A \to a\alpha \mid (s, \alpha) \in \delta(s, a, A)\}$$

  このとき $x \in \Sigma^*$ に対して

  $$M \text{ が } x \text{ を受理} \iff M_1 \text{ が } x \text{ を受理} \iff x \in L(G)$$

cfg と pda の等価性 (2/2)

証明 (2/2)

- cfg G から pda M を構成

  G を Greibach 標準形 $G_1 = (V, \Sigma, P, S)$ に直す

  $M = (\{s\}, \Sigma, V, \delta, s, S)$, ただし

  $$(s, \alpha) \in \delta(s, a, A) \iff (A \to a\alpha) \in P$$

  このとき $x \in \Sigma^*$ に対して

  $$x \in L(G) \iff M \text{ が } x \text{ を受理}$$

■

注意

注意

L_1, L_2 が Σ 上の正規言語のとき

- $L_1 \cup L_2$, $L_1 \cap L_2$, $\Sigma^* \setminus L_1$, $L_1 L_2$, L_1^* も正規言語 (第 3 回の定理 2)

L_1, L_2 が Σ 上の文脈自由言語のとき

- $L_1 \cup L_2$, $L_1 L_2$, L_1^* も文脈自由言語 (第 4 回の定理 1)
- $L_1 \cap L_2$, $\Sigma^* \setminus L_1$ は文脈自由言語とは限らない

  $L_1 \cap L_2$ の例: 第 5 回 p. 34 の例
  $\Sigma^* \setminus L_1$ の例: 第 5 回の例題 3, 今回の例題 4, 定理 2