第 15 講 DPDA と NPDA

林　恒俊

PDA のあらまし

- PDA は FSA よりスタックが追加されている分だけ構成要素が多い。したがって PDA の制御器は追加されたスタックを制御する機能が要求されるためより複雑になっている。
- FSA の制御は現状態と入力テープ上の記号をパラメータと考えれば充分であった。PDA の制御ではこれに加えてスタック上の記号も入力パラメータとして扱わなければならない。
- FSA では入力テープが計算ステップ毎に自動的に前進するため制御出力は次状態だけを考えればよかった。しかし PDA では次状態に加えて入力テープの前進制御とスタックの取扱いを行わなければならない。PDA の遷移はこれらの考察に基づいて定義される。

DPDA の遷移関数

- DPDA の遷移関数は

  $$\delta : (q, t, s) \to (q', i, d)$$

  として定義される。すなわち入力パラメータは (q, t, s) であり出力値は (q', i, d) である。これらの値について以下で説明する。
- q と q' はともに Q の要素であり
  - q は現状態を
  - q' は次状態を

  表している。
- t と s はともにテープアルファベットの記号で

  - t は入力テープ上でヘッドが読んでいる記号を
  - s は作業テープ上でヘッドが読んでいる記号を

  示している。

- i は入力テープ操作集合 $\{\circ, +\}$ の要素で
  - 要素 $\circ$ はテープを操作しない (no operation)
  - 要素 $+$ はヘッドを 1 区分前進させる

  を代表する。この出力値により状態遷移後の入力テープ動作が決定する。

- d は作業テープ操作集合 $\{\circ, \mathrm{pop}\} \cup \{\mathrm{push}\ \sigma \mid \sigma \in \Sigma\}$ の要素で状態遷移後の作業テープ動作を指定する。
  - 要素 $\circ$ はテープを操作しない (no operation)
  - 要素 pop は
    - テープの先頭の記号を取去り ($\langle$ を書込む)
    - ヘッドを 1 区分前進させる

    ヘッドはスタックの残りの記号列の先頭を参照する
  - push σ は
    - ヘッドを 1 区分後退させ
    - 記号 σ を書込む

    ヘッドはこの最後に書込まれた記号を参照する

  ヘッドは常にスタックの記号列の先頭を参照する。

- DPDA の遷移関数の引数は現状態、入力テープ上の記号、作業テープ上の記号であり、次状態、入力テープ操作および作業テープ操作が結果として得られる。遷移関数を表現する状態遷移表はこれらの要素を順に並べて構成される。

考察

上記のように本講議では PDA の定義に計算機科学に向いたモデルを採用している。しかし教科書によってはかなり異なるモデルを採用している場合もある。例えば

- 入力記号集合とスタック記号集合を互いに独立させる
- 状態遷移関数の入力記号集合に空列を追加し入力テープ制御を行わない (空遷移はテープを進めないことと同義である)
- スタック操作の代わりにテープ操作とテープ書込み文字を与える

などである。言語に関して最終的に得られる結果は採用するモデルに依存しない。

DPDA の動作の形式的定義

- DPDA の動作の定式的定義は FSA の動作の形式的定義と全く同様に与えられる。FSA の動作状態が現状態と入力テープ構成の対で代表されるのに対してより構造が複雑な DPDA の動作状態は現状態、入力テープ構成、作業テープ構成の 3 つ組で表示される。すなわち DPDA $M = (Q, \Sigma, \delta, q_0, F)$ の入力記号列 $x \in \Sigma^*$ についての動作状態は $\mathcal{C} = (q, [p, x], [1, y])$ として与えられる。ただし
  - $q \in Q$ は現状態
  - $[p, x]$ は入力テープ構成
  - $[1, y]$ は作業テープ構成でヘッドは常にテープの先頭 (スタックの頂上) を指している
- M がとりうるすべての動作状態の集合を $C(M)$ で表示する。
- M が入力記号列 x に対して動作を開始する初期動作状態 $\mathcal{C}_0 \in C(M)$

は

$$\mathcal{C}_0 = (q_0, [1, x], [1, \Lambda])$$

である。

DPDA の 1 ステップ動作定義

- 動作状態 $\mathcal{C}_1, \mathcal{C}_2 \in C(M)$ について

$$\mathcal{C}_1 = (q_1, [p_1, x], [1, y]),$$
$$\mathcal{C}_2 = (q_2, [p_2, x], [1, y'])$$

とする。$\mathcal{C}_1, \mathcal{C}_2$ の間に次の条件が成立する場合 $\mathcal{C}_1 \vdash_M \mathcal{C}_2$ と表示し $\mathcal{C}_1$ から $\mathcal{C}_2$ に 1 ステップ進むという。

- **遷移関数**
  - $(q_2, i, d) = \delta(q_1, x(p_1), y(1))$
- **入力テープ操作**
  - $i = \circ \ \wedge \ p_1 = p_2$ または
  - $i = + \ \wedge \ p_1 + 1 = p_2$
- **作業テープ操作**
  - $d = \circ \ \wedge \ y = y'$ または
  - $d = \text{push } a \ \wedge \ ay = y'$ または
  - $d = \text{pop} \ \wedge \ y = y(1)y'$ または
  - $d = \text{pop} \ \wedge \ y = y' = \Lambda$

  最後の項目はスタックが空のときに pop 操作を行ってもスタックは空のままかわらないということである。
- FSA と同様に $\vdash_M$ から $\vdash_M^*$ を定義する。
- 状態遷移関数に次状態が定義されていないなどで動作状態が未定義になれば機械は停止する。

DPDA による言語の定義

- 入力 x について M が次のような動作状態列

  $$(q_0, [1, x], [1, \Lambda]) \vdash_M^* (f, [|x| + 1, x], [1, \Lambda]) \quad \text{ただし} \quad f \in F$$

  をもてば M は x を受理するという。すなわち

  - 初期状態、入力テープの先頭、空スタックから開始し
  - 入力テープをすべて読み
  - スタックを空にした上で
  - 終了状態のいずれかに到達して停止

  した場合である。受理しない場合は入力を拒否するという。

- M が定義する言語 $L(M)$ は

  $$L(M) = \{x \mid M \text{ は } x \text{ を受理する }\}$$

  である。

DPDA の例

- $(((())())())$ のように括弧が対になって出現するような言語を考える。アルファベットを $\{l, r\}$ とする。左括弧 (を l で右括弧) を r で代表する。

- この言語を受理する DPDA $M = (\{q_0, q_1, q_2\}, \{l, r\}, \delta, q_0, \{q_1\})$ の遷移関数 δ は次の表で定義される。

q	t	s	q'	i	d
q_0	l	$\rangle$	q_0	$+$	push l
q_0	l	l	q_0	$+$	push l
q_0	r	l	q_0	$+$	pop
q_0	$\rangle$	$\rangle$	q_1	$\circ$	$\circ$
q_0	$\rangle$	l	q_2	$\circ$	$\circ$
q_0	r	$\rangle$	q_2	$\circ$	$\circ$

  入力が正しい場合には q_1 へ到達し誤っている場合には q_2 に到達する。

- このDPDAが入力記号列 $llrlrr$ を与えられた場合、次のような計算を行い受理する。

$$\begin{aligned}(q_0,[1,llrlrr],[1,\Lambda]) &\vdash_M (q_0,[2,llrlrr],[1,l])\\ &\vdash_M (q_0,[3,llrlrr],[1,ll])\\ &\vdash_M (q_0,[4,llrlrr],[1,l])\\ &\vdash_M (q_0,[5,llrlrr],[1,ll])\\ &\vdash_M (q_0,[6,llrlrr],[1,l])\\ &\vdash_M (q_0,[7,llrlrr],[1,\Lambda])\\ &\vdash_M (q_1,[7,llrlrr],[1,\Lambda])\end{aligned}$$

- lrr を入力すると

$$\begin{aligned}(q_0,[1,lrr],[1,\Lambda]) &\vdash_M (q_0,[2,lrr],[1,l])\\ &\vdash_M (q_0,[3,lrr],[1,\Lambda])\\ &\vdash_M (q_2,[3,lrr],[1,\Lambda])\end{aligned}$$

  となって受理しない。

課題　a^nb^n 型の言語を受理するDPDAを構成せよ。

DPDAとDFA

- つぎのDPDA $M_e=(\{q_0,q_1\},\{0,1\},\delta,q_0,\{q_0\})$ を考える。ただし δ は

q	t	s	q'	i	d
q_0	0	$\rangle$	q_0	$+$	$\circ$
q_0	1	$\rangle$	q_1	$+$	$\circ$
q_1	0	$\rangle$	q_1	$+$	$\circ$
q_1	1	$\rangle$	q_0	$+$	$\circ$

  とする。

- 機械 M_e は次のように1が偶数個含まれる記号列を受理する。

$$\begin{aligned}(q_0,[1,1001],[1,\Lambda]) &\vdash_{M_e} (q_1,[2,1001],[1,\Lambda])\\ &\vdash_{M_e} (q_1,[3,1001],[1,\Lambda])\\ &\vdash_{M_e} (q_1,[4,1001],[1,\Lambda])\\ &\vdash_{M_e} (q_0,[5,1001],[1,\Lambda])\end{aligned}$$

- M_e は状態遷移関数の出力でスタック操作を一切行わないため動作は例えば次の有限状態機械 $M'_e = (\{q_0, q_1\}, \{0, 1\}, \delta', q_0, \{q_0\})$ のそれと等価になると考えられる。ここで δ' は

	0	1
q_0	q_0	q_1
q_1	q_1	q_0

  である。

- 与えられた DFA に対して q, t, q' は DFA の遷移表を流用、s は常に $\rangle$、i は常に $+$、d は常に $\circ$ として DPDA の遷移表を構成すると DFA と同等動作をする DPDA が作成できる。

- DFA (あるいは FSA) が受理する言語はそれを受理する DPDA がかならず存在する。すなわち

$$\mathcal{L}_{\mathbf{RL}} = \mathcal{L}_{\mathbf{FSA}} = \mathcal{L}_{\mathbf{DFA}} \subseteq \mathcal{L}_{\mathbf{DPDA}}$$

NPDA のあらまし

- NPDA と DPDA は DFA と NFA ほど互いに異なっていない。遷移について引数と値を構成する要素は全く同一であり引数に重複が認められかどうかのみが異なっている。

- 遷移関係 Δ は

$$\begin{aligned}\Delta = \{(q, t, s, q', i, d) \mid\ & q, q' \in Q, \\ & s, t \in \Sigma_T, \\ & i \in \{\circ, +\}, \\ & d \in \{\circ, \mathrm{pop}\} \cup \{\mathrm{push}\ \sigma \mid \sigma \in \Sigma\}\}\end{aligned}$$

  あるいは

$$\Delta \subseteq Q \times \Sigma_T \times \Sigma_T \times Q \times \{\circ, +\} \times (\{\circ, \mathrm{pop}\} \cup \{\mathrm{push}\ \sigma \mid \sigma \in \Sigma\})$$

  として定義される。

- この (q, t, s, q', i, d) の各要素は DPDA に準じて解釈される。

- NPDA の動作は NFA の動作に準じるものと考えればよい。動作状態や初期動作状態は DPDA と同様に定義される。NPDA の状態遷移表で
  - ある (q, t, s) に対して複数の次動作が定義されていれば
  - 動作状態の列が複数の列に分岐するものとし
  - 動作状態列は全体で木構造を構成する
  - 木構造のいずれかの末端に受理動作状態が 1 個でもあれば
  - 全体で入力記号列を受理したものとする
- NPDA の定義する言語はその NPDA が受理する記号列を要素とする集合である。

DPDA が受理する言語と NPDA が受理する言語

- FSA では前述したように $\mathcal{L}_{\mathbf{FSA}} = \mathcal{L}_{\mathbf{DFA}} = \mathcal{L}_{\mathbf{NFA}} = \mathcal{L}_{\mathbf{RL}}$ である。
- しかし PDA では $\mathcal{L}_{\mathbf{DPDA}}$ と $\mathcal{L}_{\mathbf{NPDA}}$ は同じではない。NPDA が受理する言語が DPDA で受理できない場合がある、DPDA の定義は NPDA の定義に含まれるため DPDA は同時に NPDA でもある。すなわち $\mathcal{L}_{\mathbf{DPDA}} \subseteq \mathcal{L}_{\mathbf{NPDA}}$ は成立する。ところが次のように $\mathcal{L}_{\mathbf{DPDA}} \supseteq \mathcal{L}_{\mathbf{NPDA}}$ は成立しない。
- 言語

$$L = \{0^i 1^i \mid i \geq 0\} \cup \{0^j 1^{2j} \mid j \geq 0\} = \{\Lambda, 01, 011, \ldots\}$$

  を受理する PDA の動作は次のように考えることができる。

  1. 入力記号列の前半で 0 が連続する間、スタックに 0 を格納する
  2. 1 が出現すると入力列が
     (a) $0^i 1^i$ の要素なら 1 と同じ数
     (b) $0^j 1^{2j}$ の要素なら 1 の 2 回に 1 回

     0 をスタックから取除く

3. 入力とスタックが同時に空になれば受理する

DPDA なら 2. の段階で動作を選択しなければならないが決定手段がない。しかし NPDA なら両方の処理を並行して行い適切な側で受理することができる。

- NPDA $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \Delta, q_0, \{q_1\})$ を次のように定義すればよい。ただし状態遷移表 Δ は

q	t	s	q'	i	d
q_0	$\rangle$	$\rangle$	q_1	$\circ$	$\circ$
q_0	0	$\rangle$	q_0	$+$	push 0
q_0	0	0	q_0	$+$	push 0
q_0	1	0	q_2	$\circ$	$\circ$
q_0	1	0	q_3	$\circ$	$\circ$
q_2	$\rangle$	$\rangle$	q_1	$\circ$	$\circ$
q_2	1	0	q_2	$+$	pop
q_3	$\rangle$	$\rangle$	q_1	$\circ$	$\circ$
q_3	1	0	q_4	$+$	$\circ$
q_4	1	0	q_3	$+$	pop

これには $(q_0, 1, 0)$ 入力に対して 2 通りの遷移が存在する。

- この NPDA は次のように動作する。
  1. 前半の状態 q_0 の部分でスタックに 0 を格納している。
  2. 入力に 1 が出現すると非決定的に状態 q_2 または q_3 に移行する。
  3. q_2 では 1 が 1 個出現する毎に 0 を 1 個スタックから取除く。
  4. q_3 では、1 が 1 個出現する毎に q_4 と交替し、q_3 に戻るたびに 0 を 1 個スタックから取除く。すなわち 1 が 2 個出現する毎に 0 を 1 個スタックから取除く。
  5. q_2 または q_3 のいずれかで適切な記号列を受理する。
- このように NPDA が DPDA より能力が高いので一般に PDA といえば NPDA をさすことが多い。

言語の DPDA 化

- アルファベット $\{0,1\}$ 上の言語 $L = \{xx^R \mid x \in \{0,1\}^*\}$ は PDA で受理することが可能である。記号列の前半部をスタックに格納しておいて後半部でスタックから読出して入力の残りと比較し同じであることを確認すればよい。スタックから読出す場合に逆順になるため都合が良い。
- しかし DPDA では受理できない。折返しの位置を決定的に知る手段がないからである。
- NPDA では問題なく受理可能である。(どのようにすればよいか考察せよ)
- 言語を少し修正して DPDA で受理することを考える。例えばアルファベット $\Sigma = \{0,1,2\}$ とし言語 $L = \{x2x^R \mid x \in \{0,1\}^*\}$ とすると DPDA で受理可能である。なぜなら折返し位置を決定的に求めることができるようになっているからである。すなわち入力記号列列に 2 が出現すれば折返しと判断できる。

課題　言語 $L_P = \{xx^R \mid x \in \{a,b\}^*\}$ を受理する NPDA を設計せよ。