第 16 講 DTM と NTM

林　恒俊

TM のあらまし

- TM の構成要素は PDA と同様である。しかし PDA では入力テープは順次読取り、作業テープはスタックとしてのみ利用することが可能であったのに対して TM ではいずれのテープも前進後退操作を行うことができる。また作業テープにはいつでも記号を書込むことが可能である。
- TM の遷移の入力パラメータは現在状態、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、テープ書込み記号および作業テープ操作が出力として得られる。

DTM の遷移関数

- DTM の状態遷移関数は $\delta : (q, a, b) \to (q', d_1, c, d_2)$ として定義される。
- q と q' はともに Q の要素であり現状態と次状態を表している。
- a, b, c はすべてテープアルファベットの要素で
  - a は入力テープ上の現在の記号
  - b は作業テープ上の現在の記号

  を示している。c は作業テープに書込まれる記号で b を書換える。
- d_1, d_2 はテープ操作集合 $\{+1, 0, -1\}$ の要素で
  - 0 はテープを操作しないこと
  - +1 はテープを 1 区分前進させること
  - -1 はテープを 1 区分後退させること

を代表する。また

- ◦ d_1 は入力テープの操作を
- ◦ d_2 は作業テープの操作を

指定する。厳密な定義は次の形式的動作定義を参照すること。

考察

PDA の場合と同様に TM の定義も教科書により多かれ少なかれ異なっている。本ノートでは他の抽象機械と同一の取扱いができるように作業テープと入力テープの両方を持つ機械を対象とした。作業テープのみを備えた TM で説明している教科書もある。この場合初期動作状態で入力記号列があらかじめ作業テープ上に記録されている。いずれにせよどの TM も能力は同等であることが証明できる。

DTM の動作

- DTM の動作状態も DPDA の動作状態と同様に
  - ◦ 現状態
  - ◦ 入力テープ構成
  - ◦ 作業テープ構成

  の 3 つ組で定義される。したがって DTM $M = (Q, \Sigma, \delta, q_0.F)$ の入力記号列 $x \in \Sigma^*$ についての動作状態は $\mathcal{C} = (q, [p, x], [w, y])$ である。

  ただし

  - ◦ $q \in Q$ は現状態
  - ◦ $[p, x]$ は入力テープ構成
  - ◦ $[w, y]$ は作業テープ構成

- 機械が入力記号列 x に対して動作を開始する初期動作状態 $\mathcal{C}_0$ は

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

である。また 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。

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

$$\mathcal{C}_1 = (q_1, [h_1, x], [k_1, t_1])$$

と

$$\mathcal{C}_2 = (q_2, [h_2, x], [k_2, t_2])$$

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。

- $a = x(h_1)$ は入力テープ上の現記号、$b = t_1(k_1)$ は作業テープ上の現記号とすると状態遷移関数の値は

$$\delta(q_1, a, b) = (q_2, d_1, c, d_2)$$

ただし $d_1, d_2 \in \{+1, 0, -1\}$ 及び $c \in \Sigma_T$

この時

1. $0 \leq h_2 \leq |x| + 1$ かつ $h_2 = h_1 + d_1$ であること
2. $|t_1| = |t_2|$ の場合 $k_2 = k_1 + d_2$ かつ

$$t_2 = t_1(1) \circ \cdots \circ t_1(k_1 - 1) \circ c \circ t_1(k_1 + 1) \circ \cdots \circ t_1(|t_1|)$$

3. $|t_1| < |t_2|$ の場合
   - $k_1 = 0$ なら $k_2 = 1 + d_2$ かつ $t_2 = c \circ t_1$
   - $k_1 = |t_1| + 1$ なら $k_2 = |t_1| + 1 + d_2$ かつ $t_2 = t_1 \circ c$
4. $|t_1| > |t_2|$ の場合
   - $c = \langle$ なら $k_2 = d_2$ かつ $t_2 = t_1(k_1 + 1) \circ \cdots \circ t_1(|t_1|)$
   - $c = \rangle$ なら $k_2 = k_1 + d_2$ かつ $t_2 = t_1(1) \circ \cdots \circ t_1(k_1 - 1)$

- DTM のヘッドの位置がテープの有効範囲外に移動した場合にには機械の動作が中断 (hang) する。
- 次動作状態が未定義の場合に DTM は停止する。このような動作状態を停止動作状態という。
- $\vdash_M$ から $\vdash_M^*$ を FSA と同様に定義する。

DTM の定義する言語

- 入力 $x \in \Sigma^*$ について M に次のような動作状態列

$$(q_0, [1, x], [1, \Lambda]) \vdash_M^* (f, [h, x], [k, s])$$

  があり、$(f, [h, x], [k, s]) = \mathcal{C}_f$ とすると $\mathcal{C}_f$ は停止動作状態でかつ $f \in F$ の時 M は x を受理するという。なお $f \notin F$ の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。

考察

TM の動作は、テープ上の記号を書換えるような操作を行う点で現実の計算機の動作とよく類似している。実は書換えすなわち代入操作は形式的証明が行いにくい対象の一つである。信頼性の高いプログラムを作成するためにはプログラムの検証は非常に重要であるにもかかわらず日常的に行われていないのはこの理由による。TM のテープ操作の証明技法がプログラムの検証に有効かもしれない。

DTM の実例

- DTM $M = (\{q_0,\, q_1,\, q_2, q_3\},\, \{0,1\},\, \delta,\, q_0,\, \{q_3\})$ の遷移関数 δ は次の表で定義される。

q	a	b	q'	d_1	c	d_2
q_0	$\rangle$	$\rangle$	q_3	0	$\rangle$	0
q_0	0	$\rangle$	q_1	$+1$	0	$+1$
q_1	0	$\rangle$	q_1	$+1$	0	$+1$
q_1	1	$\rangle$	q_2	0	$\rangle$	-1
q_2	1	0	q_2	$+1$	$\rangle$	-1
q_2	$\rangle$	$\langle$	q_3	0	$\langle$	$+1$

- この DTM が受理する言語は $\{0^n1^n \mid n \geq 0\}$ である。作業テープをスタックとして利用している。確かめてみよ。

- この DTM が入力記号列 0011 を与えられた場合次のような計算を行い、受理する。

$$\begin{aligned}
(q_0, [1, 0011], [1, \Lambda]) &\vdash_M (q_1, [2, 0011], [2, 0]) \\
&\vdash_M (q_1, [3, 0011], [3, 00]) \\
&\vdash_M (q_2, [3, 0011], [2, 00]) \\
&\vdash_M (q_2, [4, 0011], [1, 0]) \\
&\vdash_M (q_2, [5, 0011], [0, \Lambda]) \\
&\vdash_M (q_3, [5, 0011], [1, \Lambda])
\end{aligned}$$

- これ例からから DPDA が受理する言語に対してそれを受理する DTM がかならず存在することが考察される。同様に NPDA が受理する言語についてそれを受理する NTM が存在することも考察される。これから

$$\mathcal{L}_{\mathbf{PDA}} \subseteq \mathcal{L}_{\mathbf{TM}}$$

  である。

- 言語 $L = \{a^nb^nc^n \mid n \geq 0\}$ を受理する TM を次のように構成することができる。

  1. 入力記号列に a が出現する間 a を作業テープに複写する

2. b が出現すると作業テープ上の a を b で書換える。
3. c が出現した時 a がすべて書換えられたかどうか確認する
4. c が出現する間 b を c で書換える
5. 入力がなくなった時 b がすべて c で書換えられたか確認する

- なおこの $a^n b^n c^n$ 型言語は後述する文脈自由言語に関するポンピング定理から文脈自由言語でないことが知られている。すなわち PDA では定義できないが TM なら定義できる言語が存在する。したがって

  $$\mathcal{L}_{\mathbf{PDA}} \subset \mathcal{L}_{\mathbf{TM}}$$

  である。

NTM

- NTM と DTM の関係は NPDA と DPDA の関係とほとんど同じである。
- 遷移関係 Δ は

  $$\begin{aligned} \Delta = \{(q, a, b, q', d_1, c, d_2) \quad | \quad & q, q' \in Q, \\ & a, b, c \in \Sigma_T, \\ & d_1, d_2 \in \{+1, 0, -1\}\} \end{aligned}$$

  あるいは

  $$\Delta \subseteq Q \times \Sigma_T \times \Sigma_T \times Q \times \{+1, 0, -1\} \times \Sigma_T \times \{+1, 0, -1\}$$

  として定義される。
- 動作の定義も NPDA の定義と平行しているので詳細は省略する。

NTM が受理する言語

- PDA の場合と異なり NTM の受理する言語のクラスは DTM の受理する言語のクラスと同じである。

  $$\mathcal{L}_{\mathbf{DTM}} = \mathcal{L}_{\mathbf{NTM}}$$

- 作業テープを利用して与えられた NTM の動作をシミュレートする DTM を構築することが可能であることが知られている。これから NTM の受理する言語は DTM の受理する言語と同一であることが説明される。
- 詳細は省略する。興味があれば計算可能性理論あるいは数学基礎論の参考書を参照するとよい。

様々な TM

- ここで説明した基本的 TM に外部記憶を追加して機能強化を図ったモデルが存在する。例えば
  - 作業テープを 2 本以上もつもの
  - 2 次元作業テープをもつもの

  などである。
- しかしこのような機能強化を行っても計算能力は基本的 TM と同等であることが知られている。なぜならどのような強化 TM でも適切にプログラミングした基本的 TM でシミュレーション可能なことが知られているからである。
- 基本的 TM でも整数を例えば 1 のならびとして表示する等の手段をとればあらゆる算術整数計算を行うことが可能である。詳細は省略するが各自考察してみよ。

判定可能性

- TM M が受理する言語を $L(M)$ とすると受理の定義から $x \in L(M)$ の入力記号列 x について M は停止する。しかし $x \notin L(M)$ の記号列について M が停止するかどうかは判らない。TM は必ず停止するとは限らないないからである。

- すべての $x \in \Sigma^*$ について TM M が停止する場合、M は言語 $L(M)$ を判定 (決定) 可能的 (decidable) に定義するという。そうでない場合を判定 (決定) 不能 (undecidable) という。
- 判定可能な言語全体を $\mathcal{L}_{\mathbf{REC}}$ で表示すると

  $$\mathcal{L}_{\mathbf{REC}} \subset \mathcal{L}_{\mathbf{TM}}$$

  である。

万能チューリング機械

- 任意の TM とその動作は状態遷移表及び $\vdash$ 関数で定義される。状態遷移表を入力記号列に符号化してその上の計算として動作を実装することができれば TM をシミュレートする TM を構築することが可能である。このような TM を**万能チューリング機械 (universal Turing machine)** と呼ぶ。
- 実際にいくつかの万能チューリング機械が提案されている。