第5章 チューリングマシン

5.1 チューリングマシン

チューリングマシン (Turing Machine) とは、__________ (Alan Turing, 1912–1954) により、計算の数学的モデルとして考えられた抽象機械である。有限個の状態を持つのは有限オートマトンと同じだが、状態のほかに、一つの記号を含むことができるセルが無限に一次元につながったテープと、テープに読み書きするためのヘッドを持ち、

- テープにあらかじめ入力が書き込まれている
- 状態と、現在ヘッドが当たっているテープ上のセルの記号で、次の動作が決まる
- 次の動作のときに、新しい状態に移るとともに、テープのヘッドが当たっているセルを新しい記号で書き換え、ヘッドを一つ右か左のセルに移動する

という点が有限オートマトンと異なる。

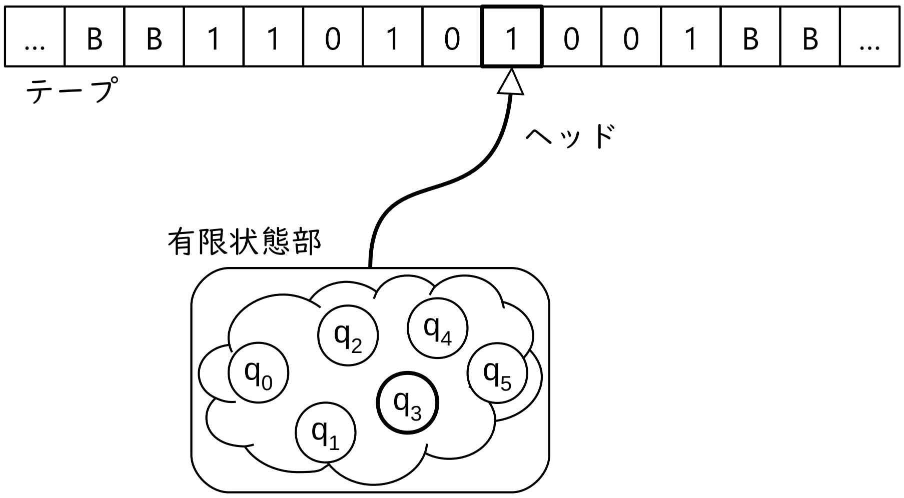

図: チューリングマシンの概念図

> 形式言語の文法としては、生成規則の形にまったく制限のない文法（タイプ-0 文法）に対応することが知られている。ただ、現実的には文脈自由文法より強力な文法が必要になることがないので、チューリングマシンを言語の解析のために使うことは、考えなくてよいだろう。

テープをメモリー、有限状態部を CPU と考えれば、現実のコンピューターのモデルとみなせる。整数や組、リストなど任意のデータをテープ上の記号列で表すことができるので、さまざまな計算をチューリングマシンで表現することができる。

チューリングマシンにも、非決定性チューリングマシンと決定性チューリングマシンがある。非決定性チューリングマシンは、次の動作にいくつかの選択肢があるチューリングマシンである。任意の非決定性チューリングマシンに対して、等価な決定性チューリングマシンを構成できることが知られている。以下では決定性チューリングマシンのみを考える。

例: 足し算をするチューリングマシン（初期状態は q_0）

	B	1
q_0	$(B, \rightarrow, q_0)$	$(1, \rightarrow, q_1)$
q_1	$(1, \rightarrow, q_2)$	$(1, \rightarrow, q_1)$
q_2	$(B, \leftarrow, q_3)$	$(1, \rightarrow, q_2)$
q_3	$(B, \leftarrow, q_3)$	$(B, \leftarrow, q_4)$
q_4		$(1, \rightarrow, \mathrm{stop})$

読み方の例: 例えば、行 q_2 と列 B の交点には、$(B, \leftarrow, q_3)$ がある、これは「テープのヘッドの位置に B を書き込んで、ヘッドを左に移動し、新しい状態を q_3 とする。」という動作を表す。

問 5.1.1 次のような初期値のテープ（ヘッドの位置は下線部）で、上の足し算をするチューリングマシンを実行せよ。

$\cdots\ B\ B\ \underline{B}\ 1\ 1\ B\ 1\ 1\ 1\ 1\ B\ B\ B\ \cdots$

例: 掛け算をするチューリングマシン（初期状態は q_0）

	B	1
q_0	$(B, \rightarrow, q_0)$	$(B, \rightarrow, q_1)$
q_1	$(B, \rightarrow, q_2)$	$(1, \rightarrow, q_1)$
q_2		$(B, \rightarrow, q_3)$
q_3	$(B, \rightarrow, q_4)$	$(1, \rightarrow, q_3)$
q_4	$(1, \leftarrow, q_5)$	$(1, \rightarrow, q_4)$
q_5	$(B, \leftarrow, q_6)$	$(1, \leftarrow, q_5)$
q_6	$(1, \leftarrow, q_{10})$	$(1, \leftarrow, q_7)$
q_7	$(1, \rightarrow, q_2)$	$(1, \leftarrow, q_7)$
q_8	$(B, \rightarrow, q_0)$	$(1, \leftarrow, q_8)$
q_9	$(B, \leftarrow, q_{11})$	$(1, \leftarrow, q_8)$
q_{10}	$(B, \leftarrow, q_9)$	$(1, \leftarrow, q_{10})$
q_{11}	$(B, \leftarrow, q_{12})$	$(B, \leftarrow, q_{11})$
q_{12}	$(B, \rightarrow, q_{12})$	$(B, \rightarrow, q_{13})$
q_{13}	$(B, \rightarrow, q_{14})$	$(B, \rightarrow, q_{13})$
q_{14}	$(B, \rightarrow, q_{15})$	$(1, \rightarrow, q_{14})$
q_{15}	$(B, \leftarrow, \mathrm{stop})$	

問 5.1.2 （結構大変）次のような初期値のテープ（ヘッドの位置は下線部）で、上の掛け算をするチューリングマシンを実行せよ。

$\cdots\ B\ B\ \underline{B}\ 1\ 1\ 1\ B\ 1\ 1\ B\ B\ B\ \cdots$

5.2 チャーチの提唱

次の事実が知られている。

- 「チューリングマシンで表現できる計算」と「__________（μ 再帰関数）で表現できる計算」と「__________(λ-calculus) で表現できる計算」はすべて等価である。（帰納的関数、ラムダ計算はいずれも、チューリングマシンと同じころに考えられた「計算」の数学的なモデルである。）

- チューリングマシン（または帰納的関数、またはラムダ計算）で表現できない「計算」を表現できる形式的な手法は、これまでに誰も見つけていない。

______________(Church's thesis) あるいは チャーチ=チューリングの提唱 (Church-Turing thesis) とは、次のような主張である。

- 「計算できる」とは、すなわち「チューリングマシン（または帰納的関数、またはラムダ計算）で表現できる」ということである

- 形式的な手法でどのように「計算」を定義しても、チューリングマシン（または…）の能力を超えられない

形式的とは「勘」とか「霊のお告げ」など他人に説明できない方法ではなく、説明可能で誰がやっても同じになる、というような意味である。この主張は証明できるようなことではないので「定理」とは呼ばず「提唱」あるいは「テーゼ」 と呼ばれる。

チューリングマシン（または…）と等価な能力をもつ計算体系を______________(Turing complete) である、という。ほとんどのプログラミング言語は、チューリングマシン（または…）をエミュレートできるので、チューリング完全である。

5.3 チューリングマシンの停止性問題

チューリングマシンは、ヘッドが当たっているテープ上の記号と有限状態部の状態から次の動作を決める、たかが有限個の規則の集まりである。だから、チューリングマシンを記号列としてエンコードすることができる。エンコードしたチューリングマシンと入力記号列の組を受け取って、その動作をエミュレートする______________________ (universal Turing machine) を構成することもできる。（コンピューターのソフトウェアで仮想コンピューターを実現するのと同じようなものである。）

すると、次のような問題を考えることができる。

> **停止性問題 (halting problem)** チューリングマシン（をエンコードした記号列）M と入力記号列 α の組を受け取って、チューリングマシン M が入力記号列 α に対して、停止するならば、$H(M, \alpha)$ が “Yes” を、停止しないならば $H(M, \alpha)$ が “No”を返すようなチューリングマシン H を定義することができるか？

これがもしできるならば、（コンピューターのプログラムはチューリングマシンと等価なので）プログラムが入力に対して停止するかどうか、を判定するプログラムが作成できることになる。（かなりの割合のバグが自動的に発見できる!!）

残念ながら、そのようなチューリングマシンを定義することは無理であることが証明されている。（**チューリングマシンの停止性の決定不能性**）

証明: もしも停止性を判定できるチューリングマシン H が存在したと仮定する。H を使って H' を次のように定義することができる。

$$\begin{cases} H(M,M) = \text{``Yes''} \text{ならば } H'(M) \text{は停止しない。} \\ H(M,M) = \text{``No''} \text{ならば } H'(M) \text{は停止する。} \end{cases}$$

$H'(H')$ が停止するかどうかを考える。停止するとすると、それは、$H(H',H')$ が "No" であることになる。しかし、これは $H'(H')$ が停止しないことを意味するので矛盾である。同様に $H'(H')$ が停止しないとすると、$H(H',H')$ が "Yes" であることになる。$H'(H')$ が停止することになり、やはり矛盾である。よって、停止性を判定できるチューリングマシン H が存在するという仮定が間違っている。（証明終わり）

チューリングマシンの停止性は、代表的な＿＿＿＿＿＿（Yes/No を決定する計算方法が存在しないことが証明されている問題）である。多くの問題の決定不能性は、チューリングマシンの停止性の決定不能性に帰着して（...が計算可能ならば、チューリングマシンの停止性も計算できる、という背理法で）証明される。