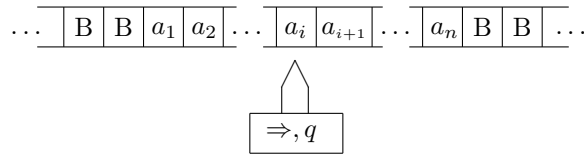


計算と論理

Jacques Garrigue, 2020 年 10 月 8 日

1 チューリング機械

チューリング機械とは、計算を行うために必要な操作を簡潔に表現したモデルである。英国の数学者 A. M. Turing が 1936 年に計算過程の定式化のために定義した。記号が書かれているテープ、それを読むヘッドと制御部の状態という三つ組からできている。テープの長さは無限だが、記号のアルファベットと制御部の状態数は有限である、さらにテープの上に書かれている記号の数も各時点では有限である。



直感的なチューリング機械の動作は三段階になっている。

1. ヘッドの下のテープの値 a_i を読む。
2. その値と制御部の状態 q に応じて、新しい値 a'_i を同じ場所に書き込み、新しい状態 q' に移る。
3. 同様に次の移動方向が決り、書き終わった後、左または右の位置 ($i-1$ または $i+1$) にヘッドを移す。

このとても単純な機械では、実はコンピュータができる全ての計算が実行できる。
チューリング機械の使う記号、取れる状態、とその状態遷移を正確に定義する。

定義 1 チューリング機械は次の 5 つ組 $M = (K, \Sigma, \delta, q_0, H)$ によって定義される。

- K : 空でない有限集合。 K の要素を **状態** という。
- Σ : 空でない有限集合 (アルファベット)。 Σ の元を **記号** という。 Σ は **空白記号 B** を含む。
- q_0 : K の要素で、 **初期状態** という。
- H : K の部分集合で、その要素を **停止状態** という。
- δ : $(K \setminus H) \times \Sigma \rightarrow \Sigma \times \{ \text{左}, \text{右} \} \times K$ なる関数で、 **遷移関数** という。

それでは機械が定義されたが、動的な状態はまだ把握されていない。そのために **時点** の概念が導入される。

定義 2 あるチューリング機械 M の **時点** はテープ、位置、状態の 3 つ組で定義される。

$$t = (T, i, q)$$

- T : $\mathbf{Z} \rightarrow \Sigma$ なる関数で、 $T(n) \neq B$ であるような n は高々有限個しかない。
- i : \mathbf{Z} の整数。
- q : K の状態。

定義 3 チューリング機械 M が一動作で時点 t から t' に移ることを $t \vdash_M t'$ と書く。

$$\delta(q, T(i)) = (a, d, q') \Rightarrow (T, i, q) \vdash_M (T', i', q')$$

- $T'(i) = a, k \neq i$ ならば $T'(k) = T(k)$
- $d = \text{右}$ ならば $i' = i + 1, d = \text{左}$ ならば $i' = i - 1$

\vdash_M の反射推移閉包を \vdash_M^* と書く。反射推移閉包とは以下の条件を満す最小の関係をいう。

継承 $t_1 \vdash_M t_2$ ならば $t_1 \vdash_M^* t_2$ (任意の t_1, t_2 について)
 反射閉包 $t \vdash_M^* t$ (任意の t について)
 推移閉包 $t_1 \vdash_M^* t_2$ かつ $t_2 \vdash_M^* t_3$ ならば $t_1 \vdash_M^* t_3$ (任意の t_1, t_2, t_3 について)

さらに、 q' は停止状態ならば、 $(T, 0, q_0) \vdash_M^* (T', n, q')$ を $T \triangleright_M (T', n, q')$ と書き、 M を T で実行した結果が (T', n, q') だという。

例題 1 2進数に1を足す機械。

テープの初期状態。ヘッドは a_0 を差す。 $a = \sum_0^k a_i \cdot 2^i$

B	a_k	...	a_1	a_0	B
---	-------	-----	-------	-------	---

$\Sigma = \{B, 0, 1\}$
 B は空記号、0 と 1 は数字

$K = \{Add, Right, Halt\}$

$q_0 = Add$

$q \backslash a$	0	1	B
Add	(1, 右, Right)	(0, 左, Add)	(1, 右, Right)
Right	(0, 右, Right)	(1, 右, Right)	(B, 左, Halt)

- 1 のときはそれを 0 に変えて繰り返し
- 0 が見つかるまで左に移動しながら繰り返す
- 見つかったら、1 に変えて右に戻る

例題 2 任意の自然数の加算 $a + b = c$

テープの初期状態 (それ以外は B)。ヘッドは M を差す。 $a = \sum_0^k a_i \cdot 2^i, b = \sum_0^l b_i \cdot 2^i$

B	a_k	...	a_1	a_0	I_0	M	b_l	...	b_0	B
---	-------	-----	-------	-------	-------	---	-------	-----	-------	---

テープの最終状態 (それ以外は B)。 $c = \sum_0^m c_i \cdot 2^i$

B	c_m	...	c_1	c_0	M	B
---	-------	-----	-------	-------	---	---

$\Sigma = \{B, 0, 1, M, I_0, I_1\}$

B は空記号、0 と 1 は数字、M は区切り、 I_0 と I_1 は現在の桁と繰り返し

$K = \{N, R, A_0, A_1, A_0^0, A_1^0, A_0^1, A_1^1, W_0, W_1, C, H\}$

Next, Read, Add (b の桁), Add (a と b の桁), Write, Clean, Halt

$q_0 = N$

$q \backslash a$	B	0	1	M	I_0	I_1
N	(a, 左, R)	(a, 右, N)	(a, 右, N)	(a, 右, N)		
R		(B, 左, A_0)	(B, 左, A_1)	(a, 左, C)		
A_i		(a, 左, A_i)	(a, 左, A_i)	(a, 左, A_i)	(a, 左, A_i^0)	(a, 左, A_i^1)
A_i^0	(a, 右, W_i)	(I_0 , 右, W_i)	(I_i , 右, W_{1-i})			
A_i^1	(I_i , 右, W_{1-i})	(I_i , 右, W_{1-i})	(I_1 , 右, W_i)			
W_i					(i, 右, N)	(i, 右, N)
C	(a, 右, H)	(a, 左, C)	(a, 左, C)		(a, 左, A_0^0)	(a, 左, A_1^0)

状態は $N \rightarrow R \rightarrow A_i \rightarrow A_i^j \rightarrow W_k \rightarrow R \rightarrow \dots \rightarrow R \rightarrow C \rightarrow A_j^0 \rightarrow W_k \rightarrow R \rightarrow C \rightarrow H$ と移行する。

練習問題 1 1. 1進数を2進数に変換する Turing 機械を定義せよ。

2. 記号 M までの部分をテープの最後にコピーする Turing 機械を定義せよ。

定義 4 ある関数 $f(x)$ が Turing 計算可能というのは、ある Turing 機械 M_f が存在し、 M_f のテープに決った形でコード化された x を書き、 M_f を実行すると、実行が必ず終わり、しかも終わったときにテープに $f(x)$ のコードが書いてあることである。

その「決ったコード」についてはいろいろ考えられるが、まず自然数は2進数で表現すればよいでしょう。組などは括弧とコンマで表現すればよいでしょう。ただし、そういうコードは必然的に各マシンのアルファベットに依存する。文字の種類が足りなければ、コードを長くすることで同じ情報が表現できる。従って、そういうコードが決っていると仮定し、 M のアルファベットでの x のをコードを \bar{x}^T と書く。

Turing 計算可能性の定義によれば、上で見た足し算を行う Turing 機械は、止まることを証明すれば、足し算が Turing 計算可能であることを示す。

2 万能チューリング機械

定義 5 ある Turing 機械 U が万能というのは、任意の Turing 機械 M と任意の入力 x について、 M の記述 \bar{M}^U と \bar{x}^U を並べたテープを U に与えると、もしも M を x に適用した結果が y であれば、 U の結果が \bar{y}^U になることである。

ここでは、二種類のものをコード化しなければならない。まず x と y は M のアルファベットで既にエンコードされていると仮定する。しかし、 U のアルファベットが M のアルファベットより小さい可能性もあるので、ここで M の各文字を一定の長さの2進数でコード化する。例えば、 M に30種類の文字があれば、 U ではそれを5倍の長さにする。

実は、 U に渡すときに2進数に変換するより、 M をあらかじめ2進数で動作する機械に変えた方が簡単。

定理 1 任意の機械 M に対し、アルファベット $\Sigma_2 = \{0, 1, B\}$ だけで動作し、 M と同じ計算をする機械 M' が構築できる。さらに、 M' のテープが一方方向にしか伸びないようにもできる。

これで、アルファベットが Σ_2 で、テープが一方方向にしか伸びない機械だけを考えればいい。

機械自体の記述はもっと難しい。ただし、紙で書けたので、テープにも書ける。初期状態、終了状態の集合、そして動作を規定する5組の集合をテープに書けばいい。

動作中にテープは以下のような形をしている。

			$q(t)$	$s(t)$	M の 5 組
...	P	...			\bar{M}^U
M の一方方向テープ			M の状態	記号	M の記述

U の動作は以下の通り

1. $(q(t), s(t))$ に合う規則を5組の中に探す。なければ終了
2. その5組に応じて $q(t+1)$ を書き込む
3. 方向を覚えながら、P を新しい文字に書き変える
4. 右か左に一個移動し、その文字を読み、P を書き込む
5. $s(t+1)$ を書き込み、1 から繰り返す

定理 2 万能 Turing 機械は構築できる。

3 判定が不可能な問題

Turing 機械によって解けない問題がある。しかも、Church-Turing Thesis を認めれば、そういう問題はどんな具体的な手続きでも解けない。

定義 6 「具体的な手続き」とは、規則の集まりと、各ステップでその規則を適用するための機構である。

Church-Turing Thesis 任意の具体的な手続きは Turing 機械によって表現できる。

具体的な手続きの定義が一般的すぎるので、この仮説を証明することはできない。しかし、今まで考えられた全ての機構は Turing 機械に還元できる。以下では、(ほとんど) 全ての計算機科学者と同様に、Church-Turing Thesis に基いて考

える。

3.1 停止問題

ある Turing 機械が停止するまでのステップ数は簡単に予測できない。永遠の止らない場合もありうる。永遠に結果を待ち続けるのは馬鹿げているので、最初から止るかどうかが判定できれば便利であろう。

Turing 機械の停止問題とは、ある機械 M とテープ T を与えられたとき、 M を T に適用した場合、実行が止まるかどうかを判定することである。

人間が判定するのは非常に難しいので、やはりコンピュータで実行できる、停止を判定する具体的な手続きが欲しい。Church-Turing Thesis を認めれば、これはすなわち、テープに \overline{M} と T を渡されたとき、有限時間で必ず返事を返す Turing 機械 H を求めることになる。 \overline{M} の記述は万能 Turing 機械のときと同じものでいい。

残念ながら、そんな Turing 機械 H は存在しえない。

定理 3 任意の Turing 機械の停止問題を判定する具体的な手続きは存在しない。

証明 もしも、停止を判定する Turing 機械 H が存在すれば、それを次のように改造する。

まず、引数として \overline{M} だけをもらい、 M が \overline{M} に適用されたとき止まるかどうかを判定する機械 H' を作る。 $H'(\overline{M}) = H(\overline{M}, \overline{M})$

次に、 H' が NO を返したときには止まるが、YES を返したときには無限ループに入る機械 H^* を作る。

最後に、 $H^*(\overline{H^*})$ が止まるかどうかを判定する、すなわち $H(\overline{H^*}, \overline{H^*}) = H'(\overline{H^*})$ を実行する。もしも YES を返せば、 $H^*(\overline{H^*})$ が止まらなくなるので、矛盾になる。NO を返せば、今度は $H^*(\overline{H^*})$ が止まってしまうので、それも矛盾。□

3.2 印字問題

定理 4 任意の機械が実行中に特定の記号 S_0 をテープに書くことがあるかどうかを判定する具体的な手続きは存在しない。

証明 S_0 をアルファベットに含まない機械 M を次のように変更する。各停止状態を変更し、停止する前に S_0 を書くようにする。

そうすると、新しい機械の印字問題は M の停止問題に当るので、判定が不可能になる。□

3.3 空テープ停止問題および一様停止問題

定理 5 任意の機械が空テープに適用されたときに停止するかどうかを判定する具体的な手続きは存在しない。

定理 6 任意の機械がテープの中身に依存せずに停止するかどうかを判定する具体的な手続きは存在しない。

証明 A, C をアルファベットに含まない機械 M を次のように変更する。実行を始める前にテープに

A	B	C
-----	-----	-----

 を書き込み (B は空記号)、左に移動しながら A を見たときそれを

A	B
-----	-----

 に書き換え、右に移動しながら C を見たときそれを

B	C
-----	-----

 に書き換える。

そうすると、新しい機械の一様停止問題は M の空テープ停止問題と一致する。□

練習問題 2 空テープ停止問題の判定不能を証明せよ。一般の停止問題に還元すればいい。