コンピュータの基礎理論

新居俊作

2023 年 4 月 17 日

1　論理回路

論理式と二進演算

P, Q を命題とするとき、その基本的な組み合わせについての真理表を、真を 1、偽を 0 として書くと以下のようになる。

P	Q	$P \wedge Q$	$P \vee Q$	$\bar{P}$	$P \oplus Q$	$\bar{Q}$
1	1	1	1	0	0	0
1	0	0	1	0	1	1
0	1	0	1	1	1	0
0	0	0	0	1	0	1

ただし $\oplus$ は排他的論理和 (XOR) と呼ばれる。

これと次の一桁の二進数の足し算を見比べる。
$X + Y = Z$ とすると：

X	Y	Z
0	0	00
1	0	01
0	1	01
1	1	10

つまり、Z の下から 1 桁目は $X \oplus Y$ で、2 桁目は $X \wedge Y$。

同様にすると、二桁の二進数の足し算は $X + Y = Z$ として、Z の各桁は以下で与えられる：

Z の下から 1 桁目： (X の 1 桁目)$\oplus$(Y の 1 桁目)

Z の下から 2 桁目： (X の 1 桁目)$\wedge$(Y の 1 桁目)
$\oplus$ $\big($(X の 2 桁目)$\oplus$(Y の 2 桁目)$\big)$

Z の下から 3 桁目： $\big($(X の 2 桁目)$\wedge$(Y の 2 桁目)$\big)$
$\oplus\Big(\big($(X の 1 桁目)$\wedge$(Y の 1 桁目)$\big)$
$\wedge$ $\big($(X の 2 桁目)$\oplus$(Y の 2 桁目)$\big)\Big)$

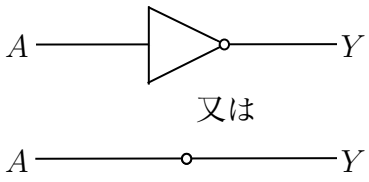
このようにして、二進数の基本的な計算は論理式で表現できる。

論理回路

電気回路を使って二進数の演算を行う機械を作りたい。その為に論理ゲートと呼ばれる素子を作り、その組み合わせで電気回路を組む。

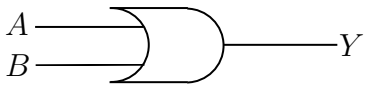
論理ゲート

NOT



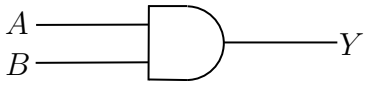
A	Y
0	1
1	0

OR



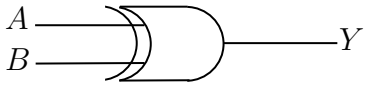
A	B	Y
0	0	0
1	0	1
0	1	1
1	1	1

AND



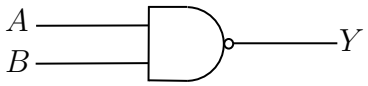
A	B	Y
0	0	0
1	0	0
0	1	0
1	1	1

XOR



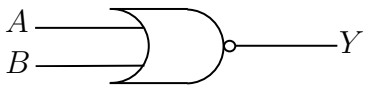
A	B	Y
0	0	0
1	0	1
0	1	1
1	1	0

NAND



A	B	Y
0	0	1
1	0	1
0	1	1
1	1	0

NOR



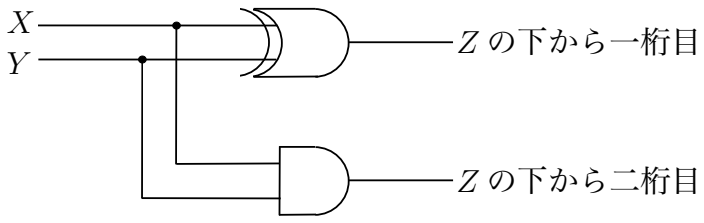
A	B	Y
0	0	1
1	0	0
0	1	0
1	1	0

論理ゲートは、初期にはリレーや真空管を用いて制作され、現在では半導体を用いて制作されている。

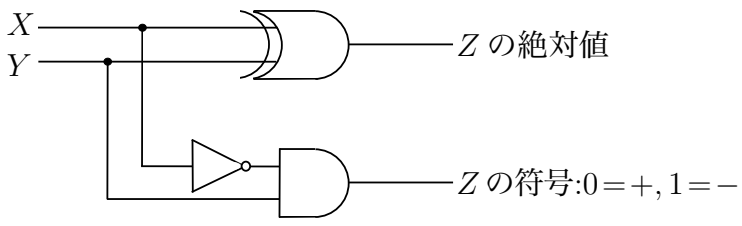
これらを用いて計算機を作る。

例　一桁の二進数の足し算機と引き算機

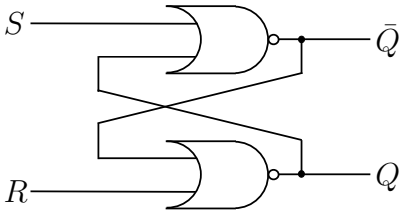
$X+Y=Z$



$X-Y=Z$



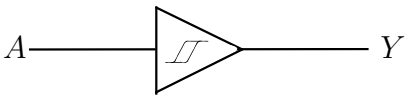
フリップフロップ

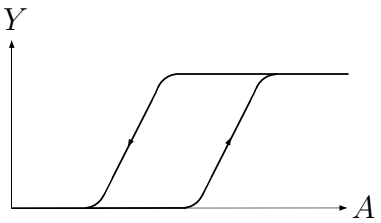


現在の Q	入力 S	入力 R	次の Q
0	0	0	0
0	0	1	0
0	1	0	1
1	0	0	1
1	0	1	0
1	1	0	1

S で Q をセットして R でリセットすることで入力を記憶させることができる。つまり簡単なメモリとして使える。

シュミットトリガ





ヒステリシス (履歴効果)

物理スイッチの不安定さを解消する。

半導体とは何か?それを用いてどう論理ゲートを作るかに着いては [幸山] 第三章等参照。

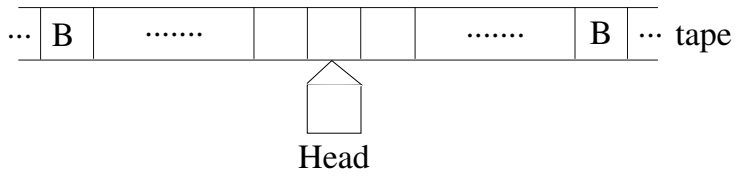
2 チューリングマシン

チューリングマシン

論理回路による計算機の限界

- 入出力の桁数があらかじめ決まっている。
  → 桁が増えると新しい計算機を作る必要がある。
- 専用機である。
  → 一つの計算機では一つの計算しかできない。

これらの問題を解決する概念：チューリングマシン。



定義 1. チューリングマシンは次のものからなる：
テープ：

- 長さは無限 (必要に応じていくらでも継ぎ足すことができる有限)。
- セルに区切られている。
- 各セルに一つの記号が書かれている。
- 書かれる記号の種類は有限。
- 有限の範囲の外のセルには B (空白を示す) が書かれている。

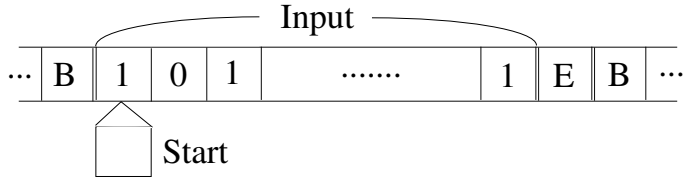
ヘッド：

- 有限な数の内部状態をとる。
- 一時刻に一つのセルの内容を読む。
- 読んだセルの内容とその時の内部状態に応じて、新しい記号をそのセルに上書きし、右または左に 1 セル動く。
- 最初にスタートするセルは決められている。

例

パリティ計算機

$0, 1$ からなる有限の長さの列について 1 の数が偶数か奇数か (パリティ) を調べる。



初期状態：

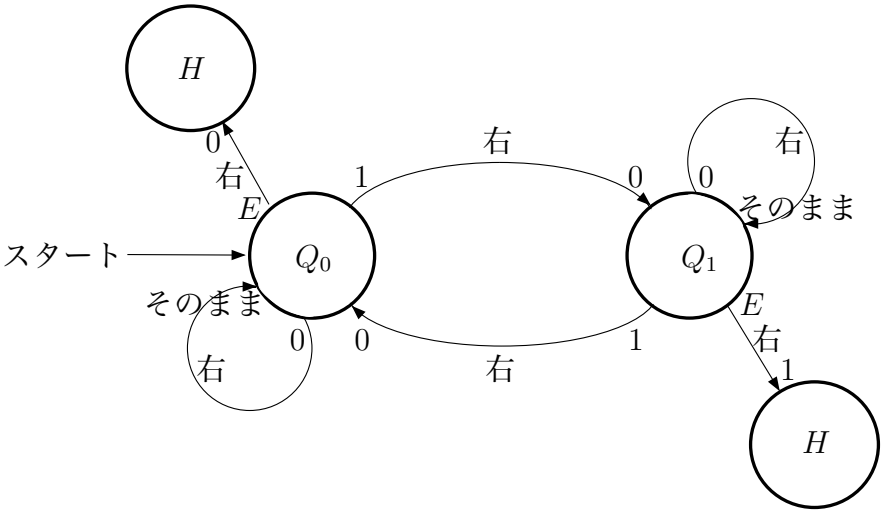
ヘッドはスタート位置 (入力の左端) にあり状態 Q_0 にある。

動作：

ヘッドは自分の位置にあるセルの内容を読み、0 なら内部状態も読んだセルの内容も変えずに、一セル右に進む。 1 なら Q_0 から Q_1、 又は Q_1 から Q_0 に内部状態を変え、セルの内容を 0 に書き変え、一セル右に進む。

読んだ内容が E ならば内部状態が Q_0 ならセルの内容を 0、Q_1 ならば 1 に書き変え、内部状態を H (holt) に変えて停止する。

これを図に表すと下のようになる。このような図を遷移図とよぶ。



この機械が停止後 E の位置に書いてあったセルの内容が 0 ならば、入力の 1 の数は偶数、1 ならば 1 の数は奇数。

パリティ計算機を論理ゲートを組み合わせて作ると入力桁が固定されてしまうが、これなら何桁でも確認できる。

問題 1. 論理ゲートを組み合わせてパリティ計算機を設計せよ。(桁数は何桁でも良い。)

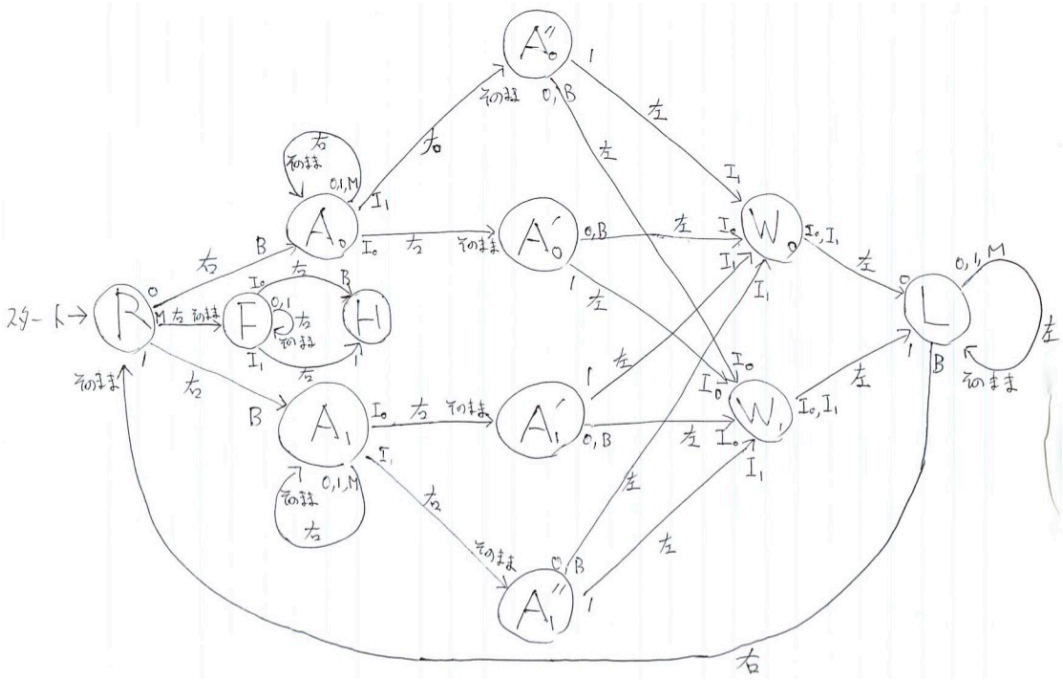
足し算機

$a = \sum_{i=0}^{k} a_i \cdot 2^i,\ b = \sum_{j=0}^{l} b_j \cdot 2^j$ (ただし $k \geq l$ とする) について $a + b$ を計算する。

···	B	a_0	a_1	···	a_k	M	I_0	b_0	b_1	···	b_1	B	···

Start

テープに書かれる記号：$B, 0, 1, M, I_0, I_1$
ヘッドの内部状態：$L, R, A_0, A_1, A'_0, A'_1, A"_0, A"_1, W_0, W_1, F, H$
ヘッドの初期状態：R
動作

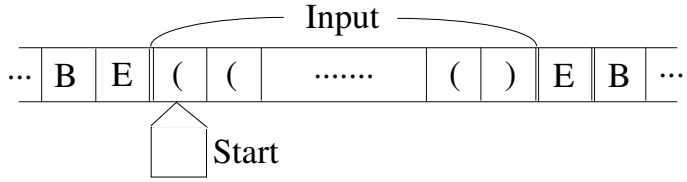


$a + b = c,\ c = \sum_{i=0}^{m} c_i \cdot 2^i$ として出力は以下のようになる：

···	B	M	c_0	c_1	···	c_m	B	···

この機械は何桁の足し算でもできる。

問題 2. 括弧検査機を設計せよ。すなわち以下の様な左括弧 (と右括弧) の列の入力に対し、左括弧と右括弧が釣り合っている (数が合っているだけではなく、ちゃんと対応がつく) ことを確認し、釣り合っていれば 1 を出力して (テープ上どこか決めたところに書き込む)、釣り合っていないならば 0 を出力して止まるチューリングマシンを設計せよ。(遷移図を描け。)



[ヒント]

スタート位置から始めて、右括弧にたどり着くまで右に移動する。右括弧のセルを読んだらそれを他の文字 (例えば X) で書き変え、左に動いていって左括弧にたどり着いたらそれをまた (X で) 書き変えて右に戻りながら右括弧を探す。

これで右の E にたどり着いたら全ての右括弧は打ち消されている。そこで再び左に移動して行き、E をにたどり着いたら全ての括弧は打ち消されているので E を 1 で書き変えて停止する。もし途中で左括弧にたどり着いたら、左括弧が余っているので、更に左に移動して行き、 E にたどり着いたら E を 0 で書き変え停止する。

右の括弧を書き換えた後で左格好を探しながら左の E にたどり着いたら右括弧が余っているので、E を 0 で書きかえ停止する。

[注意]

少なくともテープの初期状態が以下の場合についてはちゃんと動作することを確認すること：

$\cdots BE(EB\cdots$, $\cdots BE)EB\cdots$, $\cdots BE()EB\cdots$, $\cdots BE)(EB\cdots$, $\cdots BE((EB\cdots,\cdots BE))BE\cdots$

問題 3. 自分のパソコンに「オートマトンシュミレーター」：

`https://lecture.ecc.u-tokyo.ac.jp/johzu/joho/Y2018/Automaton/Automaton/AutoSim.jar`

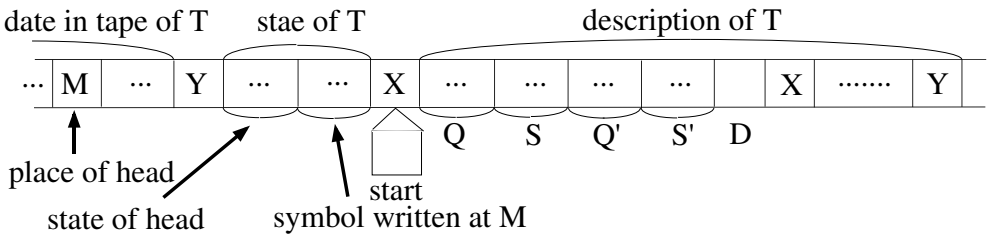
をダウンロードし、色々なチューリングマシンを作って動作を確認してみる。使い方は以下の Webpage 参照：

`https://lecture.ecc.u-tokyo.ac.jp/johzu/joho/Y2018/Automaton/Automaton/`

万能チューリングマシン

パリティ計算機や足し算機は専用マシンである。これに対し、任意のチューリングマシンの動作を再現できるチューリングマシンが存在する。これが万能チューリングマシンである。

万能チューリングマシンへの入力は以下のようになる。これによってチューリングマシン T をプログラムする。



T のヘッドは Q の状態でテープから S を読むと、Q' の状態になってテープに S' を書いて D の方向 (右 or 左) に 1 セル移動する。

T のテープに書かれる記号とヘッドの状態は有限個なのでどちらも二進数で表しておく。

動作

1. T のヘッドの状態 (the state of the head) と M に書かれた記号 (the symbol written at M) が T の記述 (the description of T) の中の Q, S と一致するものを見つける為に左から一文字づつ比較する。

2. 見つけた Q, S に続く Q', S' を一文字づつ T の状態 (state of T) の部分に描き込み、D の内容を M に書き込む。

3. 右に戻って、T の状態の部分の S' を読んで M のあった場所にそれを書き、M のところにあった D の内容に従って右又は左に M を書き、その内容を T の状態の S の上に書く。

チューリングマシンの限界 – 停止問題 –

定理 1. 任意のチューリングマシンが停止するかどうかを判定するチューリングマシンは存在しない。

証明. 背理法による。

もしその様なチューリングマシン D が存在したとする。すなわち D にあるチューリングマシン T の記述と T に対する入力テープの内容の記述を書いたテープを入力すると、T がその入力で停止するならば D は 0 を出力して停止し、T が停止しないならば 1 を出力して停止するとする。

次に新しいチューリングマシン E を次の様に作る。

E には T の記述のみを書いたテープを入力として与える。E は T の記述を T の入力領域にコピーしたあと、D と同様に動作する。

更にチューリングマシン Z を次の様に作る。

E と同じ入力に対し E が 0 を出力する場合は Z は無限ループに入って停止しない。E が 1 を出力するならば 1 を出力して停止する。

さて、Z に Z の記述を入力するとどうなるか？

Z が停止する $\Longleftrightarrow$ E は 1 を出力する $\Longleftrightarrow$ Z は停止しない。

Z が停止しない $\Longleftrightarrow$ E は 0 を出力する $\Longleftrightarrow$ Z は停止する。

これは矛盾である。よってこの様な D は存在しない。 □

3 計算可能性と計算量理論

計算可能性

万能チューリングマシンに読ませるテープの内容は、長さ有限の有限種類の文字列からなるので、その総数は可算個である。従って、チューリングマシンで計算できる実数の数は可算個である。これを計算可能実数とよぶ。(実数全体は非可算個ある。)

例

代数的数 (整数係数の代数方程式の根になる実数) は計算可能である。また、π, e 等の具体的な計算手続きが与えられている数は計算可能である。

問題 4. 代数的数が可算個であることを証明せよ。

同様に、計算可能な関数を計算可能関数と呼ぶ。

例

万能チューリングマシンに読ませるチューリングマシンの記述に番号を付けて、n 番目のチューリングマシンが停止するならば $f(n) = 0$、停止しないならば $f(n) = 1$ とする関数は計算不可能である。

計算可能実数と計算可能関数の範囲で数学を研究する分野を計算可能数学とよぶ。

計算量理論

原理的に計算可能な問題であっても、計算に非常に長い時間がかかってしまう問題は、実用上は実質的に計算不可能である。この基準として、計算に指数時間かかる問題は難しいとされる。

ねずみ算

正月に 1 つがい (2 匹) のネズミが子を雌雄 6 匹づつ 12 匹産んで、2 月にこの 7 つがいが各々雌雄 12 匹づつ産んで 98 匹になって、と繰り返して行くと、12 月には何匹になるか？(塵劫記より) *

この様に指数的に増大するものは最初は小さくても直ぐに大きくなる。

最初に与えるデータの大きさに対し、計算にかかる時間が、データの大きさの指数に従って長くなるとき、計算に指数時間かかる、と言われる。指数時間かかる問題は、データが小さい時はさほど時間をかけずに計算できても、少しデータが大きくなると、とたんに実用的な時間内では計算が終わらなくなる。

計算にデータの大きさの多項式で表される時間しかかからない場合、多項式時間で計算できると言われる。多項式時間で計算できる問題は、十分大きなコンピュータを用意すれば、実用上意味のある時間内で計算できるとされている。

例

与えられた自然数の素因数分解の計算は、その自然数の大きさに対し指数時間かかる (暗号理論の基礎)。

しかし、Shor は 1994 年に、量子コンピュータでは多項式時間で計算できることを示した [Shor]。従って、現代の暗号は量子コンピュータを使えば実用上意味のある時間内で破られるかもしれない。

参考文献

[幸山] 幸山直人「計算機理論入門 〜コンピュータを設計しよう〜」
富山大学理学部数学教室、2003 年度スーパーサイエンススクールテキスト、インターネット公開版

[ファインマン] R. ファインマン 述, A. ヘイ, R. アレン 記, 原 康夫, 中山 健, 松田 和典 訳
「ファインマン計算機科学」
岩波書店 (1999 年)

[Shor] P.W. Shor
”Algorithms for quantum computation: discrete logarithms and factoring”

*(答) 276 億 8257 万 4402 匹

Proceedings 35th Annual Symposium on Foundations of Computer Science (1994). IEEE Comput. Soc. Press: 124–134.
doi:10.1109/sfcs.1994.365700. ISBN 0818665807.