公開講座
コンピュータサイエンスアドベンチャー
～理論計算機科学はこんなに面白い～

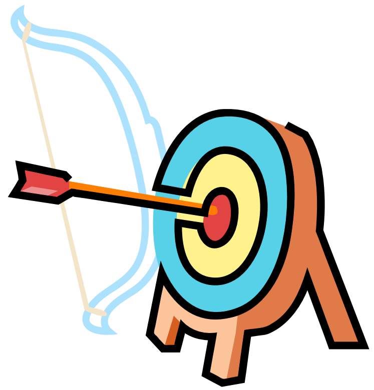
計算論

情報工学科
犬塚信博

この講座の目的

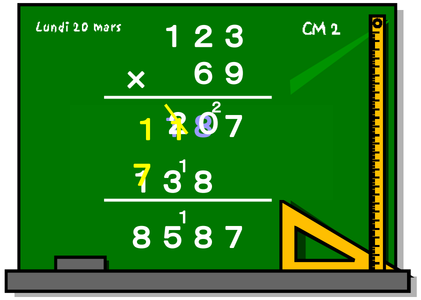
計算する、ということについて考え、計算機の原理を確認します。

- 計算をするというのは、どういうことか。
- 計算をする機械。
- 計算できないこともあるということ。
- 従来とは異なる計算の仕組み。



計算するというのはどういうことか？

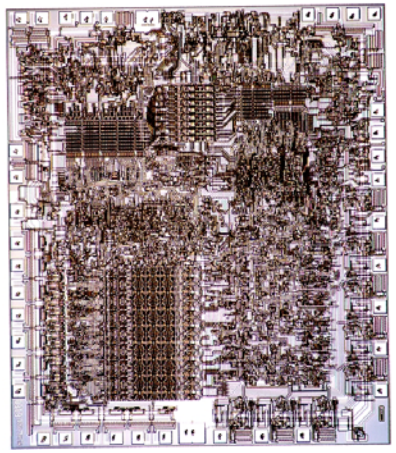
- 筆算
- 暗算
- 計算尺
- 電卓



コンピュータ

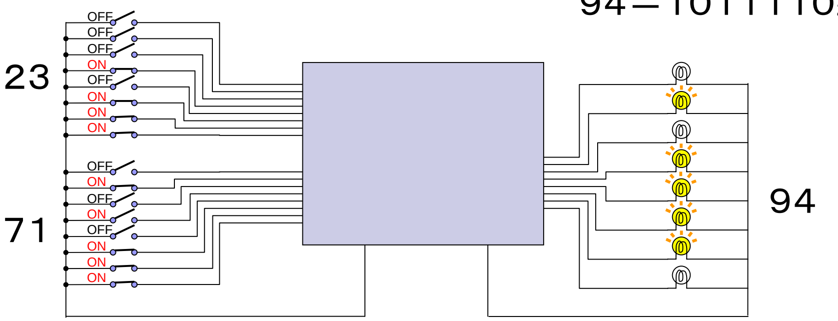
- 計算機は機械仕掛け。
- 複雑な電子部品でできている。
- しかし、その原理は比較的シンプル。

8ビットCPUチップの内部顕微鏡写真

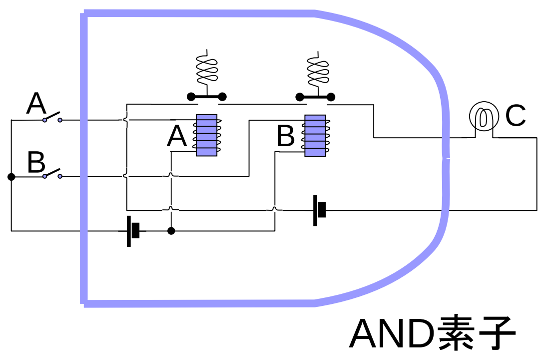


計算する機械：　～加算回路～

$23 = \quad 10111_2$
$71 = 1000111_2$
$94 = 1011110_2$

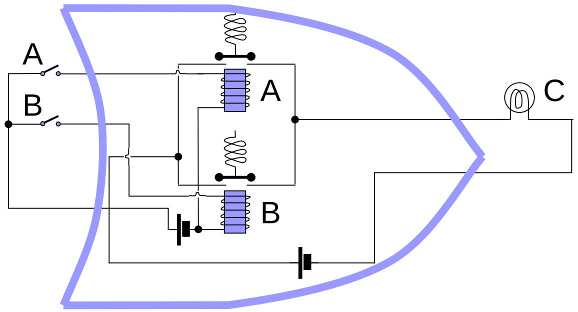


計算の基本部品（1）



A	B	C
OFF	OFF	OFF
OFF	ON	OFF
ON	OFF	OFF
ON	ON	ON

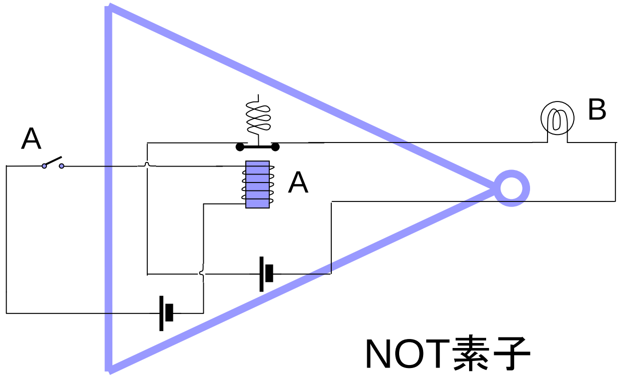
計算の基本部品(2)



OR素子

A	B	C
OFF	OFF	OFF
OFF	ON	ON
ON	OFF	ON
ON	ON	ON

計算の基本部品(3)

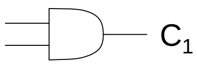


NOT素子

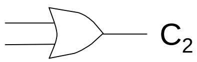
A	B
OFF	ON
ON	OFF

3つの基本論理素子

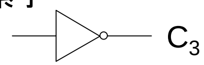
AND素子



OR素子



NOT素子



AND素子

A_1	B_1	C_1
0	0	0
0	1	0
1	0	0
1	1	1

NOT素子

A_3	C_3
0	1
1	0

OR素子

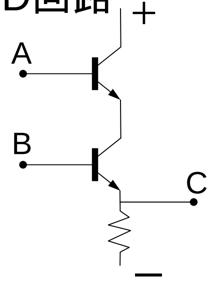
A_2	B_2	C_2
0	0	0
0	1	1
1	0	1
1	1	1

トランジスタによる論理素子回路

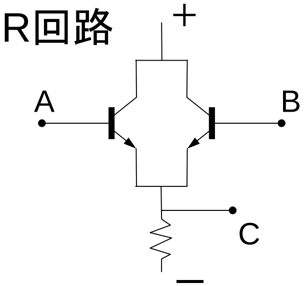
トランジスター

論理素子をつめたIC

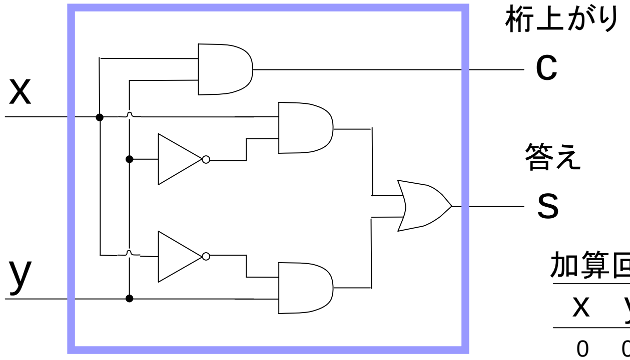
AND回路



OR回路



加算回路　2進数1桁の足し算



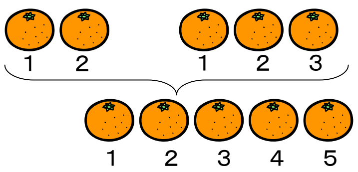
加算回路

x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

計算のための論理素子と論理回路まとめ

- AND, ORなどは、計算ための基本回路。
- これらは連動するスイッチの組み合わせでつくることができる。
- AND,ORなどを組合わせると、足し算などの計算をすることができる。

２＋３＝５ とはどういうことか。



２＋３が５であるとは、
「２」まで数えながらミカンを並べ、その後「３」まで数えながらミカンを並べなさい。
そうすると、並んだミカンを順に数えると最後に「５」が発せられる。

ということ。

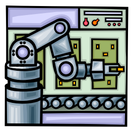
しかし、実際は。

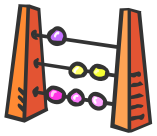
- ２＋３＝？　･･･自動的に5が頭に浮かぶ。
- １３４×６９＝？　･･･浮かんでこないので、
  筆算、または電卓。
- ミカンの意味を考えずに、機械的に計算する。

- 計算とは、機械的な記号の操作のこと。
- 機械的記号操作が、意味にかなっていることは、どこかで保証（証明）されなければならない（されている）

計算について

- 計算は、意味とは別に機械的操作である。
- 計算の意味は、数の性質からきちんと定められる。
- 機械的操作としての計算が正しいかどうかは、その意味と同じ結果になるかどうかである。
- 機械的操作なので、機械（計算機）にもできる。



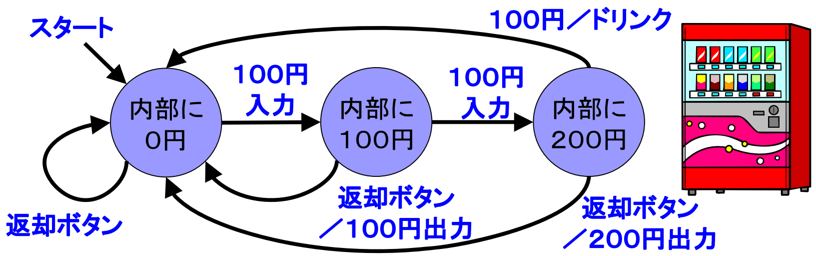


実際の計算機では、

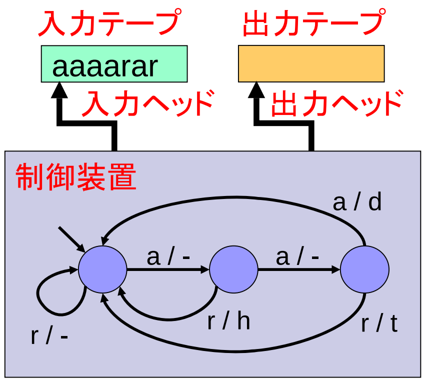
- 論理素子の単純な組み合わせ以外にいくつかの仕組みが必要。
  - 数を記憶しおくしくみ＝レジスタ
  - 時間順序にしたがって順に機械を動かすしくみ
    ＝クロックと順序機械
  - 機械の状態を覚えておくしかけ

状態機械＝自動販売機の原理

- 自販機は、内部に「すごろく」風の制御装置を想像すると分かりやすい
- 各 は、機械の内部状態。
- a／b は、その状態でaを入力するbを出力。

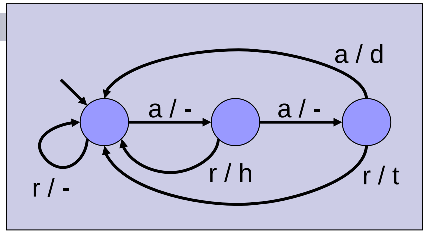


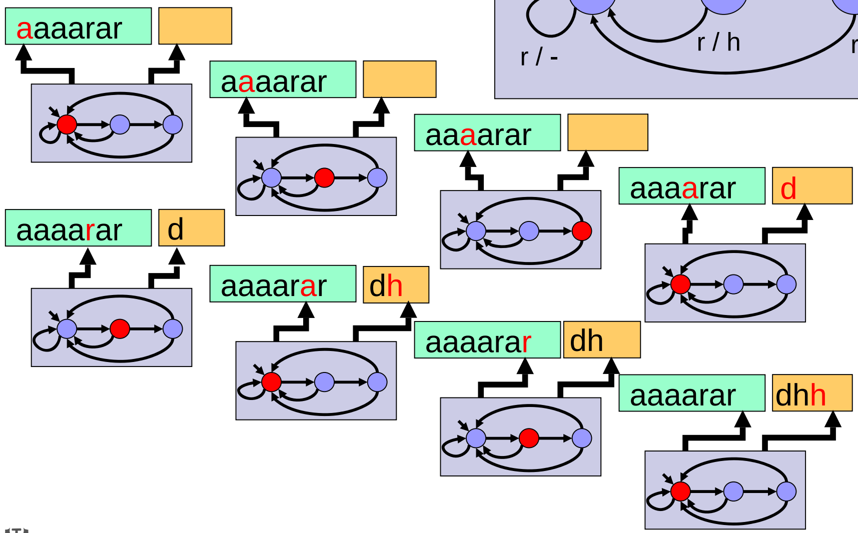
状態機械（オートマトン）



- スタートの状態から、入力にしたがって状態を遷移。
- 入力を1つ読むと、入力ヘッドを1つ右へ。
- 出力は出力テープに。1つ書くと、出力ヘッドは右へ。

状態機械の動作

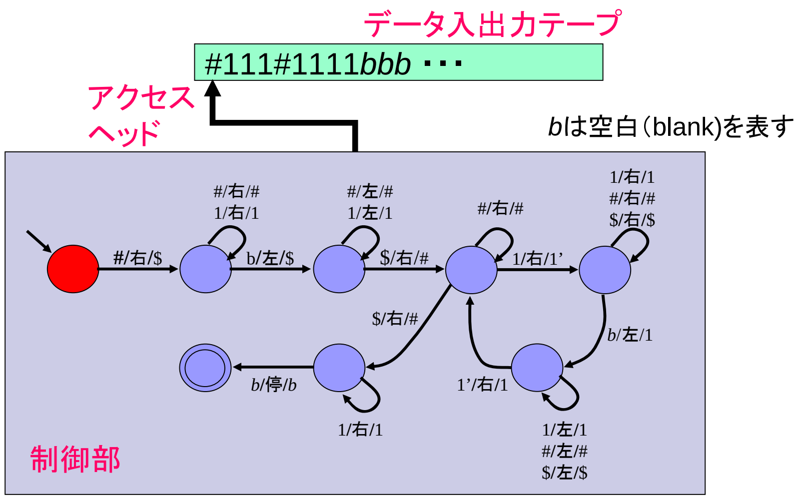




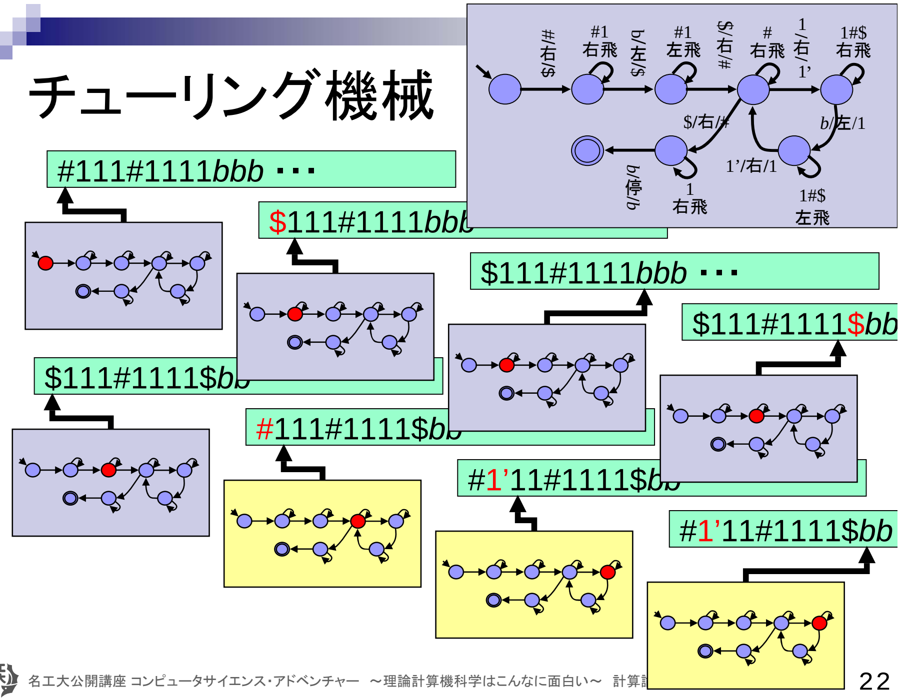
チューリング機械

- 状態機械を記号操作のために拡張。
- 入力と出力に分けず、1本のテープを持つ。
- テープは無限に長く、作業用にも使える。
- ヘッドは、左右に動くことができる。
- 1回の動作は、
  - テープから記号を読む
  - テープに記号を書く
  - ヘッドを左右に1つ動かす（動かさなくてもよい）
  - 状態を遷す

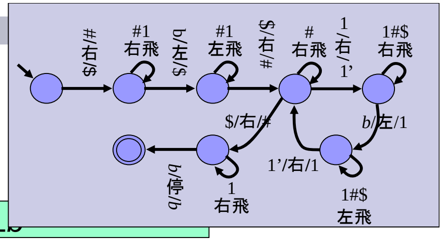
記号操作機械＝チューリング機械

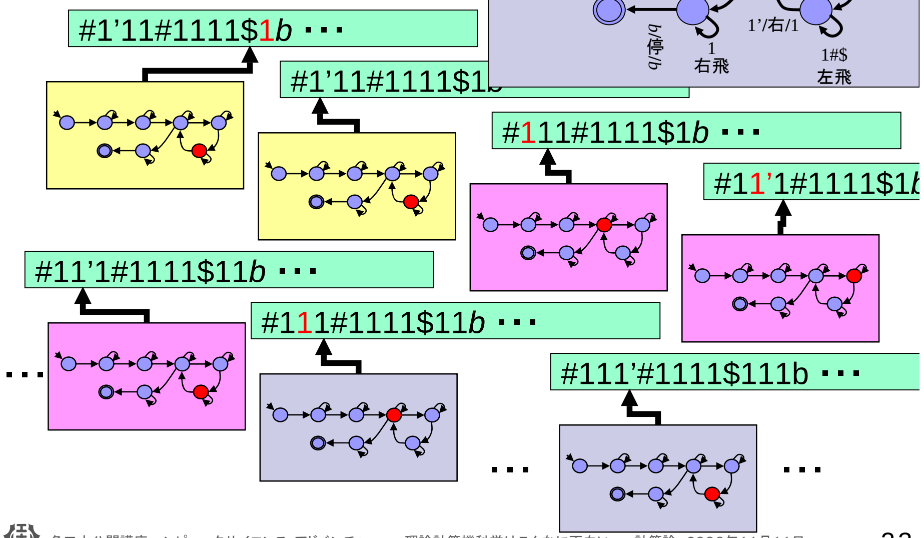


チューリング機械

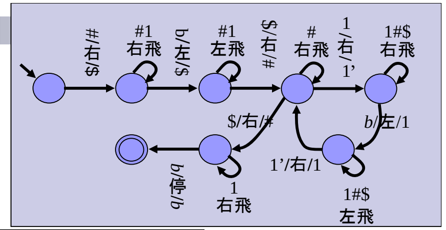


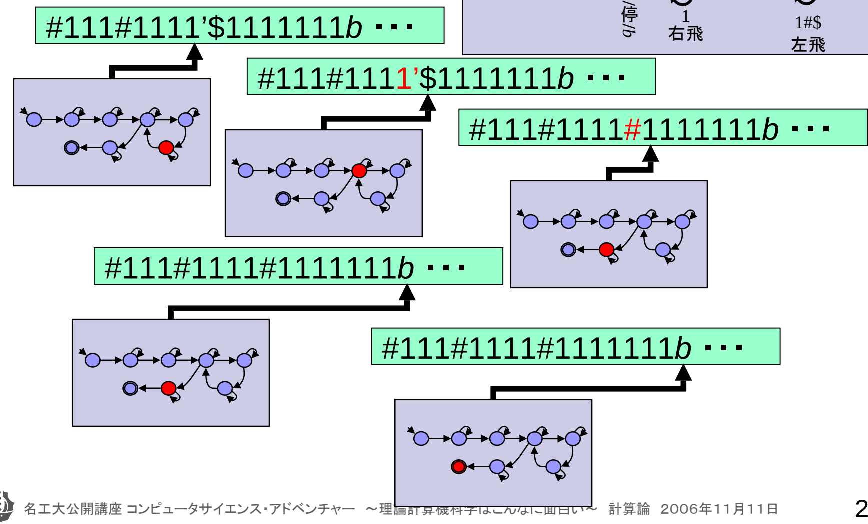
チューリング機械





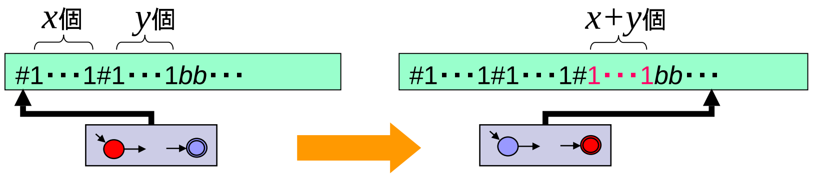
チューリング機械



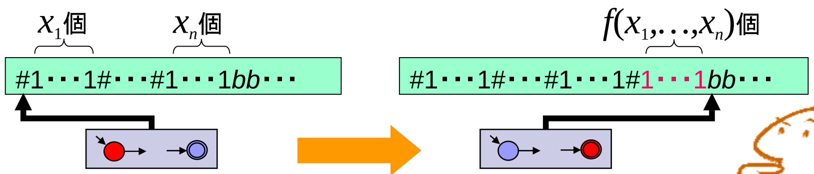


チューリング機械による計算

- x個の1が x を表すと考えれば、これは足し算を計算している。



- 次のような動作をする機械があったら、これは関数 f を計算していると思うことにしよう。



チューリング機械による計算まとめ

- チューリング機械は、いたって単純な機械であり、用意に実現できそうだ。
- チューリング機械は、予め制御部に仕込んだ順序機械にしたがってテープの記号を書き換えてゆく。
- 記号で数を表しておけば、数についての計算をするチューリング機械をつくることができる。

チューリング機械と計算可能性

- 計算することは、機械的な操作であるということはわかった。
- しかし、機械が変わればできることは変わるのではないか。

A･チューリング
1912-54

- 20世紀のはじめに、計算する機械（しくみ）が多数考案され、比較された。

チューリング機械と計算可能性 つづき

研究された計算のしくみ

- ルールによって記号を書き換えるしくみ （句構造文法、他）
- 明らかに計算方法がある関数とその合成（帰納的関数、他）
- もちろん、各種プログラミング言語（JAVA, C, FORTRAN, …）

分かったこと:

- これらのどれか1つで計算できることは、他のどの方法を使っても算できる。
- 恐らく、妥当な機能を備えたどんな計算の仕組みでも、計算できる範囲は同じだ。＝チャーチ-チューリングの提唱

←これに反する計算の仕組みは見つかっていない。

チューリング機械と計算可能性 つづき（2）

だから、

- 計算できるとは、

そのことを実現するチューリング機械があること
＝そのことを実現するCプログラムがあること
＝そのことを実現するJavaプログラムがあること
などなど

- では、計算できないことはあるのだろうか？

その前に

チューリング機械と普通のコンピュータはどこが違うか？

通常のコンピュータとチューリング機械の対応

- 入出力装置 ： 入出力テープ
- 記憶装置 ： 入出力テープ（いくらでも使える）と 制御部の状態（つくり付け）

いくらでも使える記憶装置が、テープか、ICメモリーかの違いは、計算のステップ数に違いがでる。

さらに本質的な違い:

チューリング機械は計算方法が、制御装置に作りつけ。
普通のコンピュータは、プログラム変更で機能が変わる。

プログラム内蔵方式

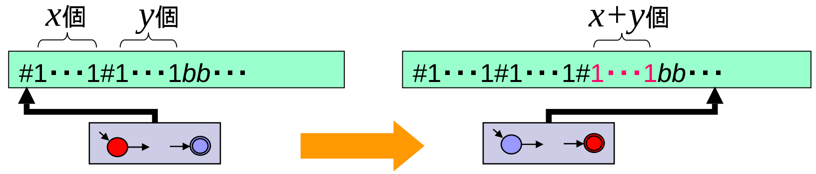
- 通常、コンピュータはプログラムをメモリーにロードし、その命令に従って動作する。
  ＝プログラム内蔵方式
  ＝ノイマン型コンピュータ

J.フォン ノイマン
1903-57

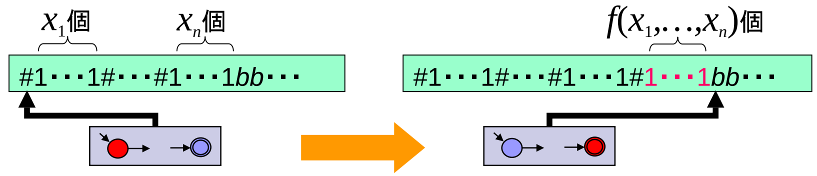
- チューリング機械をプログラム内蔵方式で作れるか？　→　できる

チューリング機械による計算

- x個の1が x を表すと考えれば、これは足し算を計算している。

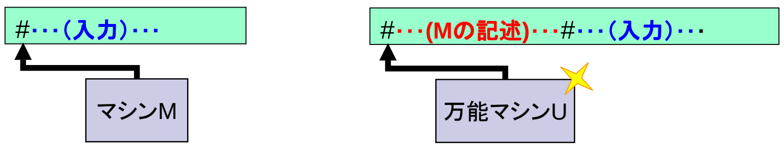


- 次のような動作をする機械があったら、これは関数 f を計算していると思うことにしよう。

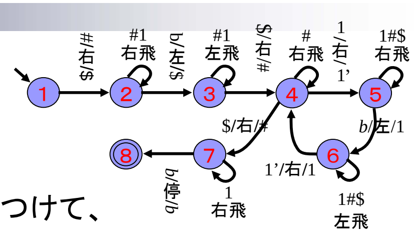


プログラム内蔵方式のチューリング機械
＝万能チューリング機械

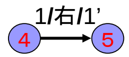
- 普通のPC同様、プログラムによっていろいろな機能を果たすチューリング機械を作りたい。
- プログラムを交換すれば、どんなチューリング機械も模倣できるようにする＝万能



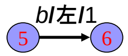
チューリング機械の記述



- 状態の記述: 状態に番号をつけて、
  q1, q11, … , q11111111　とする。
- 記号の記述:
  #: s1, $:s11, 1:s111, 1':s1111, b:s11111
- 矢印の記述:



: q1111s111Rq11111s1111



: q11111s11111Lq111111s111

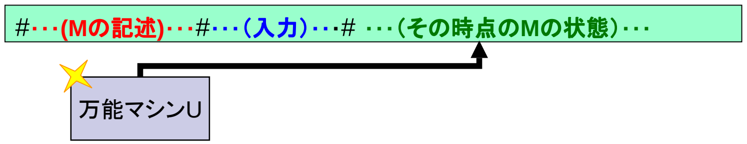
- チューリング機械の記述:
  q1 $ q11111111 $ q1s1Rq11s11 ! q11s1Rq11s1 ! ...
  スタート　ゴール　1つ目の矢印　2つ目

万能チューリング機械の動作

万能マシンUは、Mの記述を読みながらMをシミュレートします。

- マシンMの最初の状態（テープの内容、書記状態、ヘッド位置）をテープに再現
- テープに再現された状態に、Mの記述から分かる矢印の先をたどって、次の状態を再現

…



万能チューリング機械まとめ

- 万能チューリング機械は、どんなチューリング機械もその記述を与えればシミュレートできる。
- だから、これ1つだけあればチューリング機械で可能な計算ならなんでも計算できる。
- 万能チューリング機械は、1つの具体的な機械である。
- 万能チューリング機械が受取る、チューリング機械の記述は、要するにプログラム。

計算できないこと

- 計算できないことがあることを示す。

  数に関する計算に絞って考える。

考え方：

次の2つについてどちらが多いか考える。

- 数についての関数（数を数に写すやり方）
- 計算できる関数

ー どちらも無限にたくさんある?!

関数の個数、計算できる関数の個数

- 関数：

  $f(x) = x + 1$, $g(x)=x^2$, $h(x)=2^x+5x$

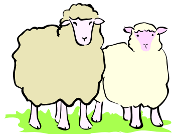
  いくらでもたくさんある。
- 計算できる関数：

  上のどれも、計算できそう。

どちらも無限にある → 無限にも程度がある。

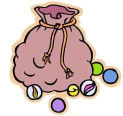
1, 2, 3, … とつづく、自然数の個数を基準に考えよう。

個数の基本は、1対1対応



- 牧場の羊の頭数と、袋の中の石が同じとは：
  羊を1つ1つ追いながら、石を1つ1つ出してゆき、
  牛と石が同時になくなること　＝羊と石が1対1対応

「個数が同じ」の基準を1対1対応とすると、



- 自然数のすべてと、偶数のすべても同じ個数
  なぜなら、

1	2	3	···	43	···
2	4	6	···	86	···

- 実は有理数（分数で書ける数）のすべても、自然数と同じ個数。
- 日本語の文章の個数も同じ個数。
  なぜ？　文章を辞書に乗せるときの順にすべて書き出して、順に1，2，3，…と対応できる。

計算できる関数の個数は、自然数と同じ個数

- 計算できるというのは、それを実現するチューリング機械があること。
- その機械を万能チューリング機械でシミュレートするためのプログラムがある。
- プログラムは、記述であるので、日本語同様、自然数と同じ個数。

自然数に関する関数は、自然数よりもたくさんある

- もし、自然数に関する関数が自然数と同じ個数なら、

1	2	3	···	i
f_1	f_2	f_3	···	f_i

  こうした1対1対応がある。

- どんな対応であっても、もしあれば矛盾することを示そう　←　対角線論法

対角線論法

各関数に1，2，3，…を与えたときの出力を表にする

		1	2	3	…	i	…
1	f_1	$f_1(1)$	$f_1(2)$	$f_1(3)$	…	$f_1(i)$	…
2	f_2	$f_2(1)$	$f_2(2)$	$f_2(3)$	…	$f_2(i)$	…
3	f_3	$f_3(1)$	$f_3(2)$	$f_3(3)$	…	$f_3(i)$	…
⋮	⋮						
i	f_i	$f_i(1)$	$f_i(2)$	$f_i(3)$	…	$f_i(i)$	…
⋮	⋮						

ここから、次の関数をつくってみる。

$f(n) = f_n(n) + 1$　：　対角線部分＋1

すると、この f は表に一切現れない

･･･全部並べたはずなのに･･･ **矛盾！**

計算できないこと まとめ

- 計算できる関数は自然数と同じ個数だけある。
- 自然数上の関数は、自然数よりたくさんある。
- 関数よりも計算できる関数は少ないのだから、計算できない関数がある。

- 具体的にはどんな関数が計算できないの？

計算できない関数

あるプログラムP
それへの入力 x → *halt* → 1 ; Pが入力 x で停止する
0 ; Pが入力 x で停止しない

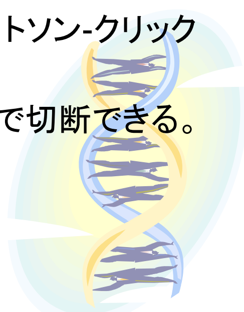
2つのプログラム P と Q → *eq* → 1 ; P と Q がまったく同じ振る舞いをする。
0 ; そうでない。

プログラム自身の性質を計算しようとすると、計算できないことが多い。

DNA

- DNAは生物の遺伝情報を保持する高分子。
- 2重螺旋構造をもつ。
- A,T,G,Cの4種類の塩基に情報が符号化され、繋がる。
- AとT,GとCが組みになって結合する＝ワトソン-クリック相補性。
- 酵素をつかって複製したり、特定の配列で切断できる。

･･･ATTCGGGCTTCGGA･･･
･･･TAAGCCCGAAGCCT･･･

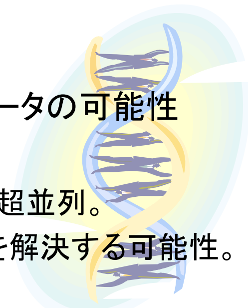


DNA計算

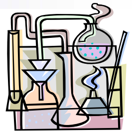
- DNAの塩基ATGCは、コンピュータの01同様、情報の符号化に利用できる。
- 生化学的反応で記号的な操作が可能：
  - DNAは、AとT、GとCが自然に結合
  - 特定の配列で切断
  - DNAを複製

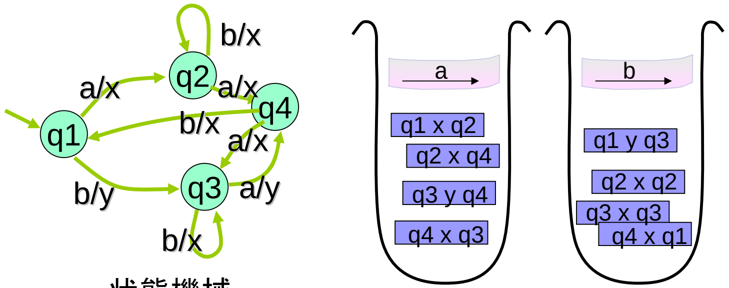
計算のために使えそう → ウェットコンピュータの可能性

- さらに
  - たくさんの分子が一度に反応するので、超並列。
  - 従来、計算量的に手に負えかった問題を解決する可能性。



DNA計算
状態機械の模倣

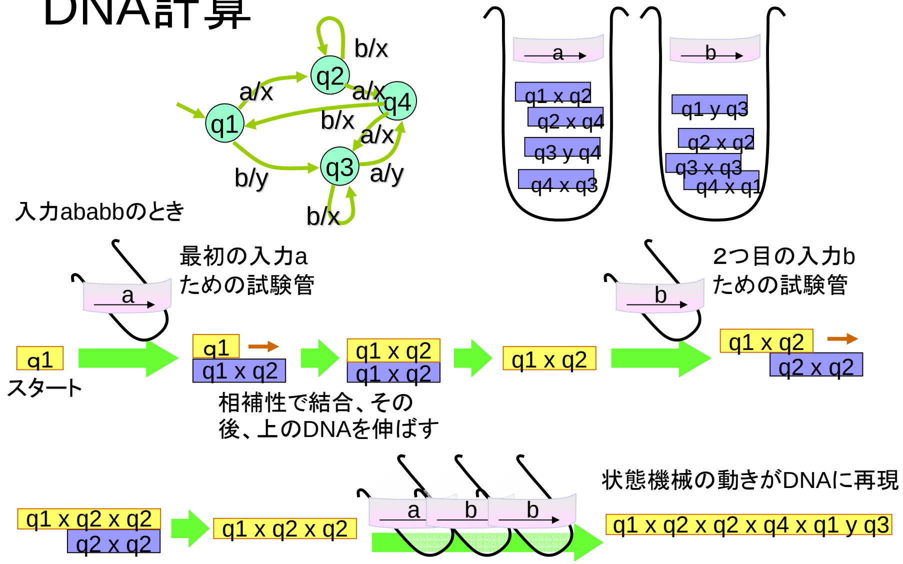




状態機械
入力：a, b

入力aのための試験管　入力bのための試験管

DNA計算



計算論のまとめ

- 「計算」の概念は機械的に操作するということ。
- 「計算」は、妥当な機能をもつ計算の仕組みならみな同じ。
- だから、性質を一般的に調べるには、チューリング機械やC言語プログラムを使えばよい。
- これらのことから、計算機の原理的な可能性を考えることができる。

計算論の応用分野

- 計算機の原理的な能力はいろいろな研究開発で応用されます。
  - 新しい原理による計算機の開発・評価
  - ロボットやネットワークシステムなど各種システムの原理的能力の評価
  - 暗号などのセキュリティシステム
  - 計算機をつかって法則を見つけるなど、高次の能力をどこまで開発できるのか。

公開講座
コンピュータサイエンスアドベンチャー
～理論計算機科学はこんなに面白い～

計算幾何学

情報工学科
本谷秀堅

ここでは話さなかった計算論の分野

- 計算の原理と、計算できること／できないことを中心に考えました。
- 計算できることを効率よく行うのは、計算量の分野（１週目のアルゴリズム）
- 計算できる問題も、原理的にどのくらい難しいのか考える分野の話はしませんでした。