

2008/07/17

情報(金 3 中村政隆先生)

実習&第3・6・7章編

ごうしょう

情報(金 3 中村 政隆先生)

実習&第3・6・7章編

情報のシケプリ（実習で扱った範囲）を作成することになったごうしょうです。とまや君と共に分担して情報のシケプリを作ることになりました。以下注意点です。

1. 講義編とは別の人が作っているので内容的に重複する可能性があります。
2. やたらマニアックになるのは必至です。内容が工学部レベルです。校閲やわからない部分は工業高専→工学部3年の石井さんに手伝ってもらいました。
3. 量も多くなりそうです。早めに読み進めることをお勧めします。
4. 「情報」の教科書の説明があまりにも飛躍しているのでごうしょうが**再構築しました**。新しい教科書を書いているようなレベルです。教科書との対応ページなども書いてあります。あ、「**6.最後に**」を**最初に読んでください**。
5. しかしこの実習編の内容も試験に出たりするので注意が必要です。
6. 実習なのでPCで実際に動かすとよいと思います。（http://lecture.ecc.u-tokyo.ac.jp/~nakamura/2008/2008_index.html）また中村先生のレジュメもわかりやすいです。CFIVEで落とせます。

7. わかりやすく説明しているつもりですが、わかりにくかったらごうしょうまで。

以下このシケプリで扱う内容…

6/6 I C トレーナー実習

6/20 WWW・メールシミュレータ実習 オートマトンシミュレータ実習 Travelling Salesman ゲーム

6/27 オートマトンシミュレータ ED21 CPU シミュレータ

7/4 HTML(文 12)

適宜教科書第3章—3.2 情報通信 第6章—6.2 計算のモデル化 第7章—コンピュータの仕組み

重要な語句は**赤をつけ**、意識すべき語句には**下線をひきました**。試験範囲外・参考程度のは**薄い青色**にしました。

また教科書に関しては「**教科書第*章**」または単に「**7.2.1**」のように章節番号のみの場合があります。「P.***」と出てきた場合も教科書を参照してください。

文体がそろってなかったり、日本語としておかしい部分が多々あると思います。勘弁してください m(__)m

1. コンピュータの仕組み（教科書第7章）	3
・ デジタルな世界(参考)	3
・ 真理値と論理関数(教科書 7.2.1) ..	3
IC トレーナー実習(6/6)	6
ED21 シミュレータ(6/27)	12
実際のコンピュータ(7.6)	17
2. WWW・メールシミュレータ(6/20)(教科書 3.2)	20
WWW シミュレーター	20
3.2.2 プロコトル	21
メールシミュレータ	
3. オートマトンシミュレータ(6/27 6.2.1)	24
6.2.1 機械的な計算モデル	24
4. Travelling Salesman ゲーム（6/27 4.3.2）	26
セールスマン問題	26
5. 補足	27
ハミング距離(o7 個別問 2)	27
イミテーションゲーム(o6 個別問 4)	28
ナイキスト周波数(o6 共通試験共通問題 1 (a))	29
6. 最後に	30
5 組シケ対よりひとこと（速読スキップ可）	30
共通試験の範囲	31
中村先生のページより	33
あとがき	34

1. コンピュータの仕組み (教科書第7章)

・デジタルな世界(参考)

アナログとデジタルの違いについてはもうよいでしょう。コンピュータの世界ではほぼすべてのデータはデジタル信号として扱われています。ところでデジタルというのは数値、という意味ですが、私たちが普段よく使っている数字(digit; 10進数が基本)とは異なり、デジタルな世界では2進数(0と1のみ)で表されています。これを実際には電圧のON/OFFで区別しています。膨大なデータであっても2進数の桁数を増やして対応しているわけです。この桁数をビット(bit)とっています。第3章のビットとは異なることに注意してください。

デジタルの世界では電圧のON/OFFは2進数の0と1に対応しています。これからの説明においては電圧のON/OFFを2進数の0と1を同値に考えます。

・真理値と論理関数(教科書7.2.1)

コンピュータは2進数で入力、計算、出力を行っていることがお分かりいただけたでしょう。2進符号表現の演算を実現する回路は、組合わせ回路と順序回路に大別されます。組合わせ回路とは計算途中を保存できないが任意の2進演算を行うことのできる関数のような回路を言います。つまり単純な計算を行います。それに対して順序回路とは計算途中を保存する機構を持ち複雑な計算を行えるようになった回路を指します。ここでは組合せ回路について説明します。では、どのような演算ができれば任意の2進演算を行えるのでしょうか？

2進数の加算、といっても10進数の加算と基本は同じです。記数法と位取りに注意しましょう。ここでは一番簡単な加算、つまり(1ビットの変数 x) + (1ビットの変数 y)について考えましょう。このときの結果を s (1ビット)とし、位が上がった時に C_{out} (1ビット)を出力することにしましょう。

x	y	s	C_{out}	10進法
0	0	0	0	0+0=0
0	1	1	0	0+1=1
1	0	1	0	1+0=1
1	1	0	1	1+1=2

(x, y は1ビットの変数 s は1ビットの結果 C_{out} は桁上げの出力)

このような関係になります。10進数表記もしたので簡単に理解できるでしょう。このように2進演算の入出力の関係を示した表を真理値表といいます。

実際の計算の時は下位の計算結果からの桁上げを考慮するので桁上げ入力 C_{in} も考慮します。

	x	y	C_{in}	s	C_{out}	10進数
①	0	0	0	0	0	0+0+0=0
②	0	0	1	1	0	0+0+1=1
③	0	1	0	1	0	0+1+0=1
④	0	1	1	0	1	0+1+1=2
⑤	1	0	0	1	0	1+0+0=1
⑥	1	0	1	0	1	1+0+1=2
⑦	1	1	0	0	1	1+1+0=2
⑧	1	1	1	1	1	1+1+1=3

(x , y は 1 ビットの変数 C_{in} は桁上げ入力 s は 1 ビットの結果 C_{out} は桁上げの出力)

2 進数の各桁の演算を前の 2 例において入力桁数を増やした形、つまり n 入力 1 出力の論理関数全種類と、その実行法を考えます。 n ビットの入力パターンは 2^n 通り (前の表を見れば自明) でありそれぞれの入力パターンについて 0 または 1 の出力が可能だから論理関数 o_i の種類 (真理表は全部で何パターンか?) は 2 の 2^n 乗となります。

それではこれらの任意の演算が行える関数は何でしょうか? それは **{AND, OR, NOT} の組み合わせ** です。(なお、説明は p.159 を見てください。また、AND, OR, NOT がどのような論理関数かは **ブール代数** の項で扱います。) このすべての論理関数を実現できる演算の組み合わせ (これらを組み合わせることですべての論理演算ができる組み合わせ) を **完備な組み合わせ** と呼び、この性質を **完備性** といいます。

また **NAND** や **NOR** は其々が単独に**完備性**を持ちます。単一の演算で完備となるため実際の論理回路ではほとんどが NAND と NOR でできています。(※制作が簡単な NAND が実際では使われることが多いようです。)

ブール代数 (7.2.2)

先ほどは組合せ回路の構成や性質をあらわすのに真理値表を用いましたが、そのほかにブール代数と MIL 記法があります。

ブール代数 は **論理回路を数式** として表わすためのもので、AND を $\cdot$ (論理積)、OR を $+$ (論理和)、NOT を $\bar{x}$ (論理否定) で表します。

AND 演算 ($\cdot$ 論理積) … 2 つの入力 x, y にたいして両方が 1 でない限り 0 を返します。つまり 2 進数 1 ビットの掛け算をするわけです。

OR 演算 ($+$ 論理和) … 2 つの入力 x, y に対して片方でも 1 であった場合は 1 を返

し、両方 0 の場合のみ 0 をかえします。つまり 2 進数 1 ビットの足し算をします。

NOT 演算 ($\bar{x}$ 論理否定) … 1 つの入力にたいして 0 ならば 1、1 ならば 0 を返します。つまり **ビットを反転** させます。

わかりにくいひとも出てくるかもしれませんが、数 A の集合と同じです。

AND を $x \cap y$ 、OR を $x \cup y$ 、で考えるとよいでしょう。

x	y	s	1 0 進法
0	0	0	$0 * 0 = 0$
0	1	0	$0 * 1 = 0$
1	0	0	$1 * 0 = 0$
1	1	1	$1 * 1 = 1$

AND 演算の真理値表

x	y	s	1 0 進法
0	0	0	$0 + 0 = 0$
0	1	1	$0 + 1 = 1$
1	0	1	$1 + 0 = 1$
1	1	1	$1 + 1 = 1$ (実際とは異なりますが、ここは説明のために便宜的にこうしてあります。なお 10 進数の計算に近い答えを足すのは半加算器)

OR 演算の真理値表

x	S
1	0
0	1

NOT 演算の真理値表

論理式(論理演算)	Boole代数式
$x \cap y$	$p \cdot q$
$x \cup y$	$p + q$
$\overline{x}$	$\overline{p}$

対応表

この他にも NAND,NOR、XOR があります。

NAND($\overline{x \cdot y}$)・**NOR**($\overline{x + y}$)…それぞれ NOT+AND NOT+OR なので詳しい説明は省きます。AND、OR の結果を反転させればよいわけです。

XOR (排他的論理和 $x \oplus y$) …基本的には OR 演算と同じです。しかし x,y がともに 1 であった場合は 0 を返します (**排他的**)。2つのビットを比較して同じならば 0 を、異なれば 1 を返します。(cf. 5. 補足 ハミング符号)

x	y	NOT	AND $x \cap y$	OR $x \cup y$	NAND $\overline{x \cap y}$	NOR $\overline{x \cup y}$	XOR $x \oplus y$
1	1	0	1	1	0	0	0
1	0	0	0	1	1	0	1
0	1	1	0	1	1	0	1
0	0	1	0	0	1	1	0

様々な論理演算の真理値表

加算標準系・乗算標準系

ブール代数表現においては、数学における他の演算と同様に同じ内容の式でも様々な表現方法があります。それらの中でいかなる演算においても唯一の表現に定まる形式を**標準系**といいます。

加算標準系…変数の総数が n 個である際に n 個の変数からなる論理積の論理和で表現された形式です。

乗算標準系…変数の総数が n 個である際に n 個の変数からなる論理和の論理積で表現された形式です。

再簡(minimal)加算標準形

命題1 $A + A \cdot B = A$

命題2 $A \cdot p + B \cdot \overline{p} = A \cdot p + B \cdot \overline{p} + A \cdot B$

例 $E = x \cdot y \cdot z + \overline{x} \cdot \overline{z} + x \cdot y \cdot \overline{z} + \overline{x} \cdot \overline{y} \cdot z + \overline{x} \cdot y \cdot \overline{z}$
 $= x \cdot y \cdot z + \overline{x} \cdot \overline{z} + x \cdot y \cdot \overline{z} + \overline{x} \cdot \overline{y} \cdot z$
 $= x \cdot y \cdot (z + \overline{z}) + \overline{x} \cdot \overline{z} + \overline{x} \cdot \overline{y} \cdot z$
 $= x \cdot y + \overline{x} \cdot \overline{z} + \overline{x} \cdot \overline{y} \cdot z$
 $= x \cdot y + \overline{x} \cdot \overline{z} + \overline{x} \cdot \overline{y} \cdot z + \overline{x} \cdot \overline{y}$
 $= x \cdot y + \overline{x} \cdot \overline{z} + \overline{x} \cdot \overline{y}$
 $(= x \cdot y + \overline{x} \cdot \overline{z} + \overline{x} \cdot \overline{y} + y \cdot \overline{z})$ (再簡加算標準形) 19

定理1は、 $A + A \cdot B = A(1 + B) = A$ ってことだね。どんなものでも、1 との OR をとると 1 になるので、 B は消えます。定理2は、 $A \cdot p + B \cdot \overline{p} = A \cdot p(1 + B) + B \cdot \overline{p}(1 + A) = A \cdot p + A \cdot p \cdot B + B \cdot \overline{p} + B \cdot p \sim \cdot A \overline{p} = A \cdot p + B \cdot \overline{p} + A \cdot B(p + \overline{p}) = A \cdot p + B \cdot \overline{p} + A \cdot B$ という変形をしてるんだね。 $1 + 0 = 1$ だから、好きなところにかけていい。1 と AND とっても変化しないから。結局、定理1は無駄な項を消すテクニック、定理2は欲しい項を取り出すテクニックということです。

- 基本積 P が E の主項であるとは、 $E+P=E$ かつ P の真部分積がその性質を持たないこと。
- 定理 任意のブール式に命題 1, 2 の変形を適用すると有限ステップで終了し、かつ E は E の主項の和になっている。

簡単に言うと主項っていうのは、ある式をできるだけ単純化した式の中にある、それぞれの項のこと。ある式はいろんな形に変形できるけど、「主項の和で表す」って決めたら、その形はひとつに決まります。この形のことを加法標準形といって、論理回路を設計する際の基本にするわけです。※ちなみに論理回路では AND と OR は対等だから、乗法標準形というのも存在します。こっちはある式を完璧に因数分解された形にもっていくわけです。

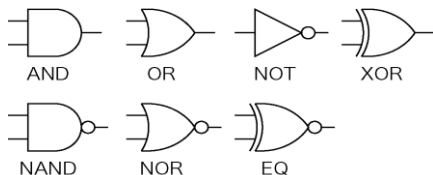
以上石井さんの解説でした。神☆正直手に負えない;;

MIL 記法 (7.2.3)

論理演算を数式として表わす方法を紹介しました。もう一つ論理演算を回路図として表わす方法があります。これを **MIL 記法** (アメリカの軍用規格の一部) といいます。以下授業用スライドを示します。

MIL 記法

- 基本的な論理演算を以下の基本素子で表現



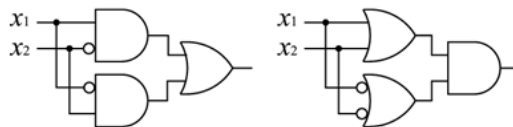
- 基本素子のあいだの結線によって論理関数を表現

20

これらの基本素子 (EQ は教科書にありませんが触れません。) のあいだを結線することで論理回路を示します。なお、NOT、NAND、NOR に見られる ○ は否定を表します。

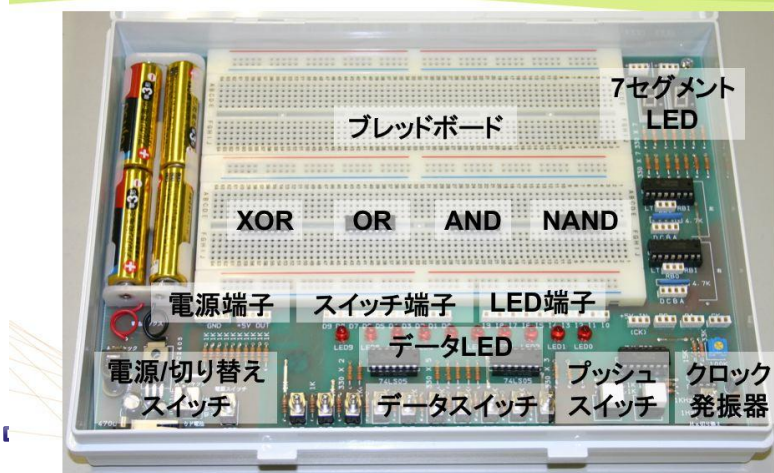
XOR の標準形による表現

- XOR (排他的論理和)
 - 2 入力と同じ値のとき 0, 異なる値のとき 1
- ブール代数による XOR の表現
 - 加算標準形 $x_1 \cdot \overline{x_2} + \overline{x_1} \cdot x_2$
 - 乗算標準形 $(x_1 + x_2) \cdot (\overline{x_1} + \overline{x_2}) =$ 展開すると...になる
- MIL 記法による XOR の表現
 - 加算標準形
 - 乗算標準形



21

IC トレーナーの構成



Copyright © the University of Tokyo

また XOR はほかの論理素子からつくるとこうなるらしいです。授業用スライドより

IC トレーナー実習 (6/6)

やっと実習に入れました。実習編をよんでわかりにくいところがあれば適宜戻ってください。

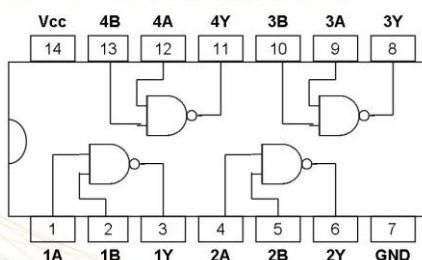
まずは IC トレーナーの各部分について説明しましょう。

- ◇ 電源端子・電源/切り替えスイッチ…この IC 群を動かすための電源部分です。安定した電源を得るために電源用 IC が入ってます。ここはあまり問題ないでしょう
- ◇ ブレッドボード…各種電子部品やジャンパ線（部品間をつなげる為の線）を差し込むことで電子回路を組むことのできる電子基盤の一種。組み立て・組み換えが簡単なので実験用につかわれる。
- ◇ スイッチ端子・データスイッチ…端子とスイッチはつながっている。スイッチは入力のための装置として使われる。計算のための数値を入力する。
- ◇ LED 端子・データ LED…LED（発光ダイオード）を光らせることにより計算結果を出力する。スイッチ・LED は **入出力装置 (I/O)** といえる。
- ◇ その他…今回は関係ありません。それではやっていきましょう。回路の接続方法・動作確認は飛ばして…

NAND ゲートの動作確認

NAND ICのピンの配置

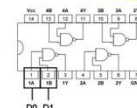
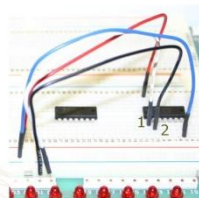
◇ NANDゲートが4つ組み込まれている



一番右にある NAND IC74LS00 の中にはこのような論理回路が入っています。なお Vcc は電源+で GND は電源-に接続します。

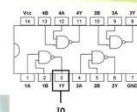
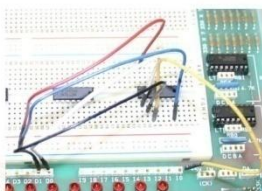
NANDゲートとスイッチ端子の接続

1. 1番ピンとD0端子を接続
2. 2番ピンとD1端子を接続



NANDゲートとLED端子の接続

◇ 3番ピンとI0端子を接続



IC と各部品を接続します。1 A と 1 B にはスイッチ D0 と D1 を、1 Y には LED I0 を接続します。

NANDゲートの動作

◇ NANDゲートに接続されている端子



◇ NANDの真理値表と、真理値と実験ボードの対応

D0	D1	I0
0	0	1
0	1	1
1	0	1
1	1	0

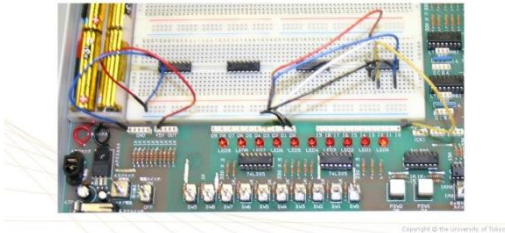
	0	1
回路	電圧低	電圧高
スイッチ	OFF (手前)	ON (奥)
LED	消灯	点灯

それでは、動作を確認します。スイッチおよび LED と端子は、SWo と Do, SW1 と D1, LEDo と Io がそれぞれつながっています。たとえばスイッチ SWo を ON にし、SW1 を OFF にすると、NAND ゲートへは D0 から 1, D1 から 0 が入力され、1 が Io に出力されますから、LEDo が点灯します。実験ボードの電源スイッチを ON にし、ス

イッチ SW₀ と SW₁ の ON, OFF を切り替えて、LED₀ が真理値表のとおり点灯もしくは消灯することを確認しましょう。

NANDゲートの実際の動作

↘ SW₀がON(1)でSW₁がOFF(0)のとき、
LED₀は点灯(1)



…まあコピペです。突っ込みどころがありません。論理演算と実際の動作がリンクしたでしょうか？

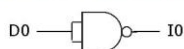
NAND ゲートの応用

NAND 演算で NOT 演算を作る

▶ NAND 演算と NOT 演算の関係

$$\overline{x} = x \cdot x$$

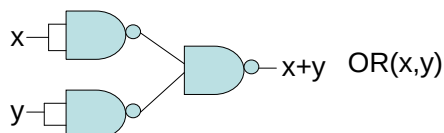
▶ NAND で作った NOT の MIL 記法による表現



NAND では NOT を作ることができます。AND 部分の 2 入力を同じにすればいいのです。そうすると、AND{0,0}=0 AND{1,1}=1 なので、反転して
NAND{0,0}=0 NAND{1,1}=1 ⇔
NOT{0}=1, NOT{1}=0 となります。

NAND ゲートで AND ・ OR 演算をする

NAND 素子からほかの論理素子を作る



22

これは実習において「必須課題」
 (<http://lecture.ecc.u-tokyo.ac.jp/~nakamura/2008/IC-trainer-exercise.html>) でした。説明を示すので、上の図において指で追っていくとわかりやすいです。

NAND ゲートで AND ゲートを作る…ひとつめの NAND では x,y 2 つの入力に対して NAND 演算を行います。ふたつめの

NAND ゲートを NOT 演算を行います。つまり、**NOT{ NAND{x,y} }** というわけです。

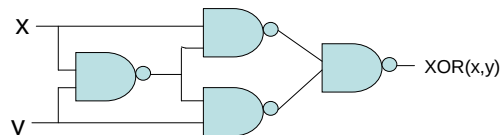
NAND ゲートで OR 演算をする…ちょっと難しいです。実際に数値をいれて順に計算していくとよいです。x,y の入力をそれぞれ反転させて NAND 演算します。段階別にして真理値表をしめします。

	入力	NOT	AND	NOT	s
x	1	0			
y	0	1	0	1	1

x,y の値を変えることでほかの場合を示すことができます。つまり **NAND{ NOT{x}, NOT{y} }** といえます。

NAND から NOT,AND,OR が作り出せました。「NAND 演算は単独で完備性を持つ」、とさきほど書きましたが、それを MIL 記法で示しました。

NAND ゲートで XOR を作る



23

選択部分です。もうわかると思いますので解説はしません。ここまで MIL 記号で示してきましたが、これをブール代数で示すとこうなるらしいです。

NANDでAND, OR, XORを作る

▶ NAND演算とAND演算, OR演算の関係

- ANDは2つ, ORは3つのNANDで作ることができる

$$\begin{aligned}x \cdot y &= \overline{\overline{x \cdot y}} = \overline{\overline{x} \cdot \overline{y}} \\x + y &= \overline{\overline{x + y}} = \overline{\overline{x} \cdot \overline{y}}\end{aligned}$$

- 各演算の真理値表は教科書162ページの表7.6参照

▶ (発展) NAND演算とXOR演算の関係

$$x \oplus y = x \cdot \overline{y} + \overline{x} \cdot y$$

- 4つのNANDゲートでXORゲートを作ることができる
(□の部分は1つのゲートの出力を2回使う)

もう意味はわかるはずです。発展的ですよ。

1ビット半加算器

それではこれらの論理 IC を組み立てることで計算をしていきましょう。すべての計算の基本である **加算器** についてやっていきます。加算器のビット数を増やしたりすることですべての計算ができます。1秒に1つカウントすることで時計になったり。実際石井さんは高専時代ロジック IC だけで時計を作られそうになって焦ったそうです。IC を 20 ~ 30 個必要なのかな? カオスになることは確実。ちょっとロジック IC がコンピュータの基本になっていることが実感できるようになったと思いますよ。

まずは1ビット半加算器。一番初めにでた真理値表を覚えていますか? **1ビット+1ビット=2ビットの足し算**です。

x	y	s	C _{out}	10進法
0	0	0	0	0+0=0
0	1	1	0	0+1=1
1	0	1	0	1+0=1
1	1	1	1	1+1=2

これをブール代数表現や論理関数表現で表すと…

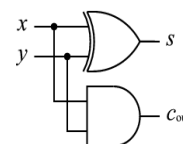
$$s = x \oplus y = \text{XOR}\{x, y\} \quad C_{out} = x \cdot y = \text{AND}\{x, y\}$$

となります。これを MIL 記号で示すとこのようになります。それを **1ビット半加算器** といいます。

1ビット半加算器

- 2つの1ビット入力x, yに対して和sと桁上げC_{out}を出力

x	y	s	C _{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



24

指で追うと合点できるでしょう。下位ビットの桁上げを考慮しないので半加算器と呼ばれます。

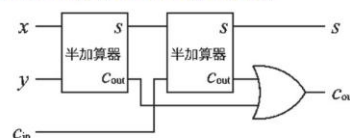
1ビット全加算器

それでは下位ビットの桁上げを考慮しましょう。

▶ 1ビット全加算器の作成

- 使用するゲート数はAND 2, XOR 2, OR 1

▶ 1ビット全加算器のMIL記法表現



▶ MIL記法と実験ボードの対応

- 入力x, y, C_{in}: スイッチ端子D0, D1, D2
- 出力s, C_{out}: LED端子I0, I1

Copyright © the university of Tokyo

真理値表:

	x	y	C _{in}	s	C _{out}	10進数
①	0	0	0	0	0	0+0+0=0
②	0	0	1	1	0	0+0+1=1
③	0	1	0	1	0	0+1+0=1
④	0	1	1	0	1	0+1+1=2

⑤	1	0	0	1	0	$1+0+0=1$
⑥	1	0	1	0	1	$1+0+1=2$
⑦	1	1	0	0	1	$1+1+0=2$
⑧	1	1	1	1	1	$1+1+1=3$

この全加算器を数珠つなぎにすることで何ケタもの数字を扱えるようになります。つまり n ビット全加算器ができてしまうわけです。

減算器 (参考 7.3.4)

加算ができることがわかりました。それでは減算はどうするのでしょうか？そのためには入力を変更します。最上位のビットを符号ビットとして固定してしまいます。そして数値の絶対値部分を 2 の補数表現をつかってしめします。まあなんのことやらわかりませんが…ようは正の数のビットを反転させると負の数にかわります。これらをそのまま加算すると引き算になっている…そうです。まあ要は **ビットを反転させることで引き算ができるのです**。

	符号 bit	絶対値	10進数
正の数	0	1 0 1	+5
負の数	1	1 1 0	-2
加算結果	0	0 1 1	+3

ALU(参考 7.3.4)

そんなこんなので足し算・引き算ができました。これらをもっと変化させると加減乗除ができるらしいです。拡張していくことで様々な計算ができるユニットを **ALU (Arithmetic and Logic Unit 算術・論理演算器)** といいます。要はコンピュータにおいて実際に計算を行う部分です。

ED21 シミュレータ (6/27)

やっと終わりました。IC トレーナー。あれを理解するにはこれだけの知識が必要だったのですね。

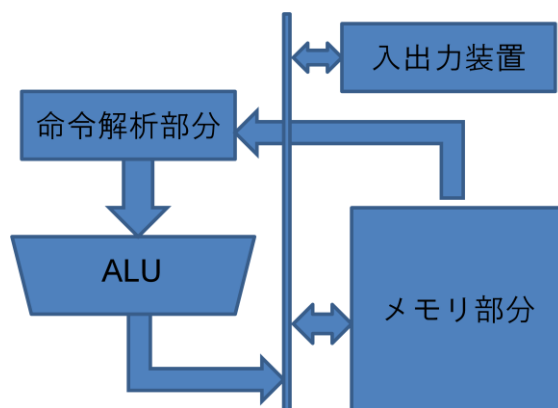
それではやっていきましょう。…このシミュレータは ECCS からでないと使えないみたいです。なお、「じょうほうの情報」(<http://www.geocities.jp/infofinf/>)に用意しておきます。

マイコン (参考)

このシミュレータは私たちが普段使っているパソコンを簡略化したものです。現在ではこれに近い構造をとっているのがマイコンと呼ばれているものです。マイコンは一定の目的のために機能が限定されたコンピュータ、と言えるでしょう。テレビのリモコンを作るのにパソコンは必要ないでしょうし…

コンピュータの基本構成 (参考 7.1.1)

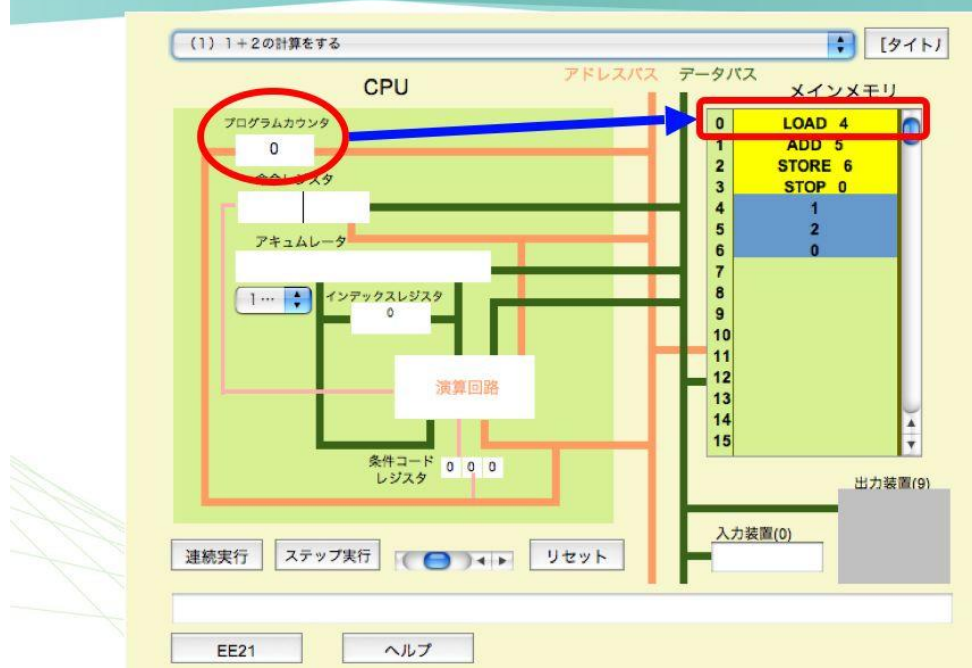
このコンピュータの構造 (アーキテクチャ) は一つのメモリの中にプログラムとデータが混在しています。これをフォンノイマン型



コンピュータと呼びます。

コンピュータはプログラムを見て計算をしていきます。そのプログラムは機械語 (0 と 1 の羅列) で書かれています。しかしこれを人が扱うことはとても難しいです。そこでこの機械語の命令を英単語に置き換えたものをアセンブリ言語と呼びます。以下の説明ではこのアセンブリ言語を使ってプログラムが書かれています。

"1+2の計算" の詳細



動作 (7.1.2 7.5.1)

まずはこのシミュレータの各部分の説明。

- 演算回路…おなじみ **ALU** などの計算を行う回路です。
- メインメモリ…プログラム (ALU がどのような計算をすべきかの命令群) やデータ (計算を行うための) を格納してあります。格納部分一つ一つには**アドレス**と呼ばれる番号が振られていて、それを指定することでメモリの内容を参照します。倉庫みたいな感じ。
- **プログラムカウンタ**…メモリの何行目の命令を実行しているかを示すカウンタです。命令が1つ終了したらカウンタが一つ増えて次に移ります。
- **命令レジスタ**…今実行すべき計算が何かを保存しておく場所です。アセンブリで**命令 | 参照すべきアドレス**の順で書かれています。
- **アキュムレータ**…ワーキングメモリというマイコンもありますね。今行っている命令でつかう数字を保存し、この数字に数値を加減していきます。
- その他…今回は関係ないみたいです。

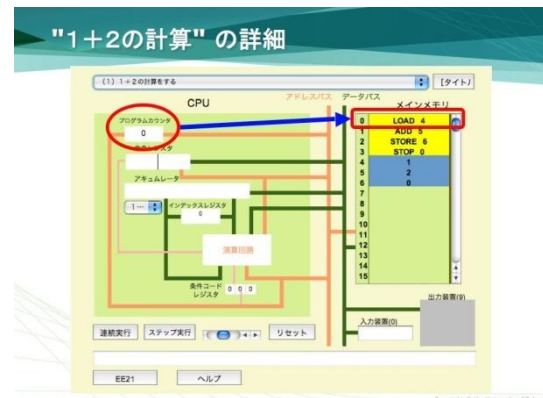
それでは実際に動かしましょう。
以下コピペです。つつこみどころがないのです。

2. "1 + 2 の計算" の詳細

"1 + 2 の計算" をするプログラムが、実際にどのように動いているのか、詳しく見てみましょう。

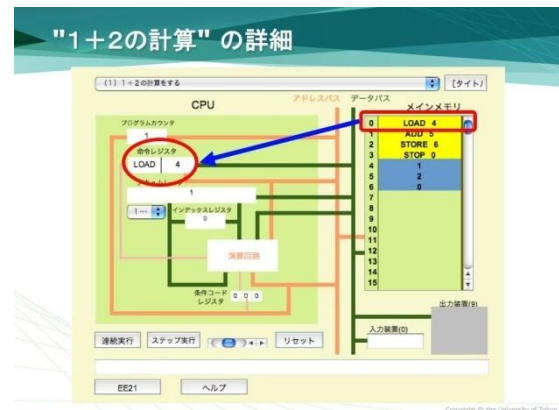
1. "プログラムカウンタ" を見て, "メインメモリ" 上の命令をとりこむ

"プログラムカウンタ" の値にしたがって, "メインメモリ" の 0 番地の命令をとりこみます。



2. "命令レジスタ" 上の命令 "LOAD 4" を実行する

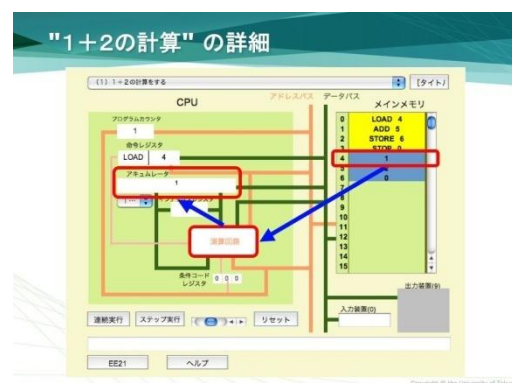
"メインメモリ" の 0 番地にあった命令 "LOAD 4" が "命令レジスタ" にとりこまれ, この "LOAD 4" を実行します。



3. "LOAD 4" の結果を"アキュムレータ" に格納する

LOAD 命令は

- LOAD n: n 番地の内容をアキュムレータにコピーするという命令です。ここでは "LOAD 4" なので, 4 番地の値である "1" を "アキュムレータ" にコピーします。

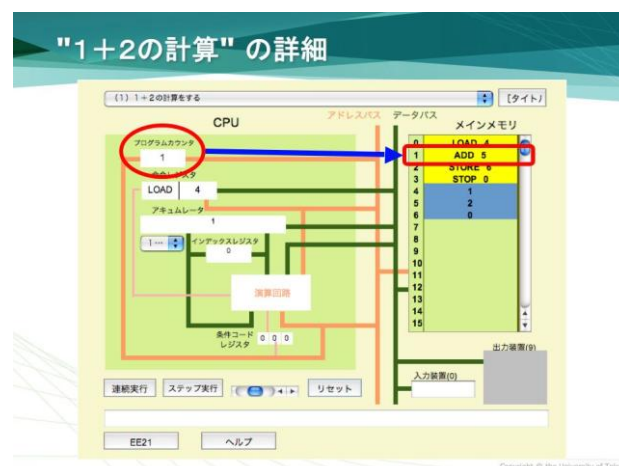


4. "プログラムカウンタ" の値を1増やす
"プログラムカウンタ" の値を1つ増やします。

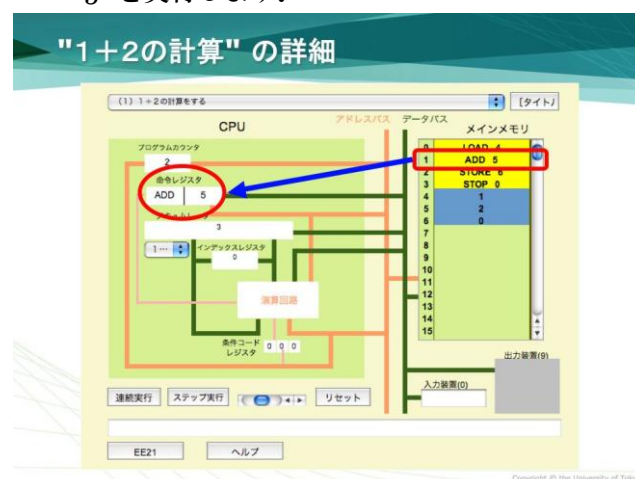


"ステップ実行"の最初のクリックで、ここまで実行が進みます。上の2と3の画像では、プログラムカウンタの値が"1"になっていますが、これは画像を取り込むための都合によるもので、本来ならここで初めて"プログラムカウンタ"の値は"1"になります。

5. "プログラムカウンタ"を見て、"メインメモリ"上の命令をとりこむ
"プログラムカウンタ"の値にしたがって、"メインメモリ"の1番地の命令をとりこみます。

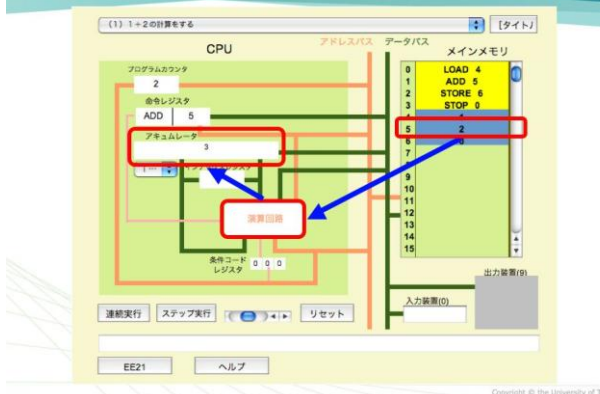


6. "命令レジスタ"上の命令"ADD 5"を実行する
"メインメモリ"の1番地にあった命令"ADD 5"が"命令レジスタ"にとりこまれ、この"ADD 5"を実行します。



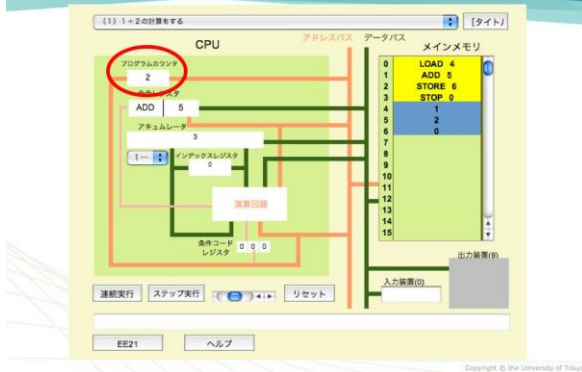
7. "ADD 5"の結果を"アキュムレータ"に格納する
ADD 命令は
• ADD n: n番地の値をアキュムレータの値に加える
という命令です。ここでは"ADD 5"なので、5番地の値である"2"を"アキュムレータ"の値である"1"に加えた値、すなわち"3"を"アキュムレータ"に格納します。

"1+2の計算"の詳細



8. "プログラムカウンタ"の値を1増やす
"プログラムカウンタ"の値を1つ増やします。

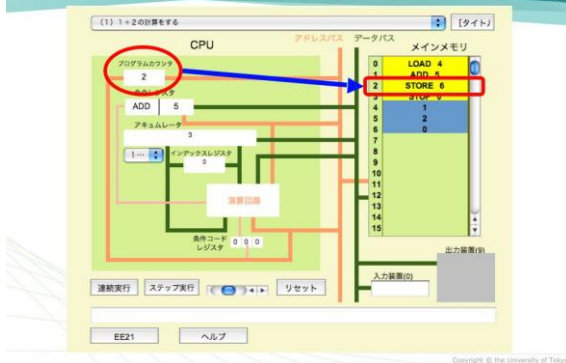
"1+2の計算"の詳細



"ステップ実行"の2回目のクリックで、こ
こまで実行が進みます。

9. "プログラムカウンタ"を見て, "メインメモ
リ"上の命令をとりこむ
"プログラムカウンタ"の値にしたがって, "メ
インメモリ"の2番地の命令をとりこみます。

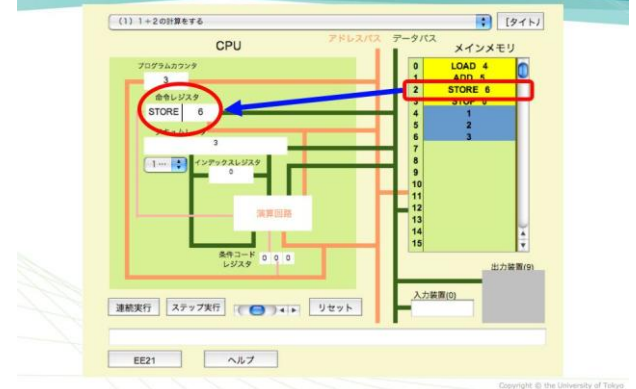
"1+2の計算"の詳細



10. "命令レジスタ"上の命令 "STORE 6" を実
行する

"メインメモリ"の2番地にあった命令
"STORE 6"が"命令レジスタ"にとりこまれ、
この"STORE 6"を実行します。

"1+2の計算"の詳細

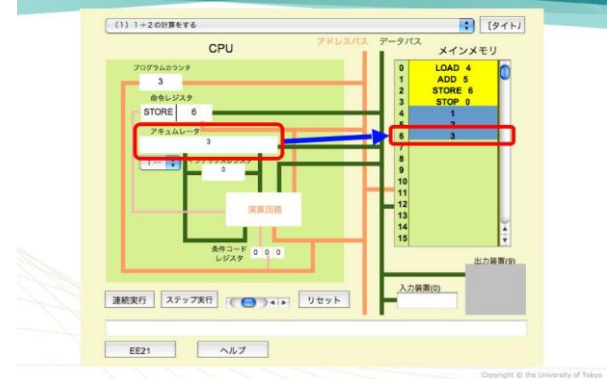


11. "STORE 6"により, メインメモリに"アキ
ュムレータ"の値を格納する

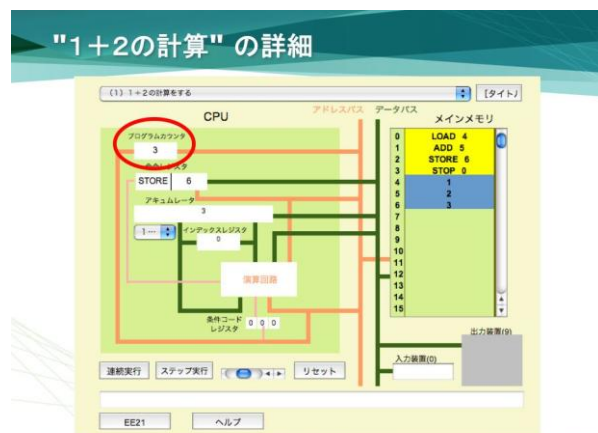
STORE 命令は

- STORE n: アキュムレータの値をn番地に格納
する
という命令です。ここでは "STORE 6" なので、
"アキュムレータ"の値"3"を6番地に書き込
みます。

"1+2の計算"の詳細



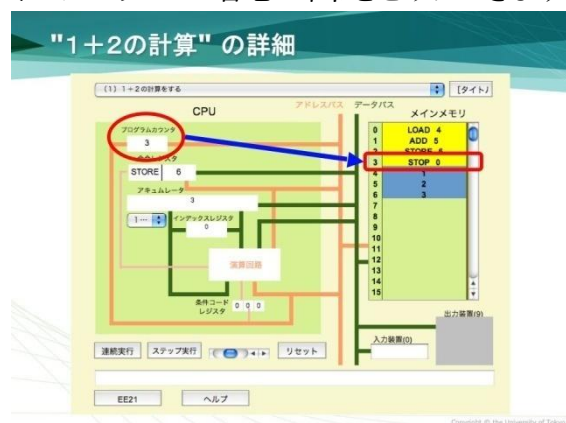
12. "プログラムカウンタ"の値を1増やす
"プログラムカウンタ"の値を1つ増やします。



"ステップ実行"の3回目のクリックで、ここまで実行が進みます。

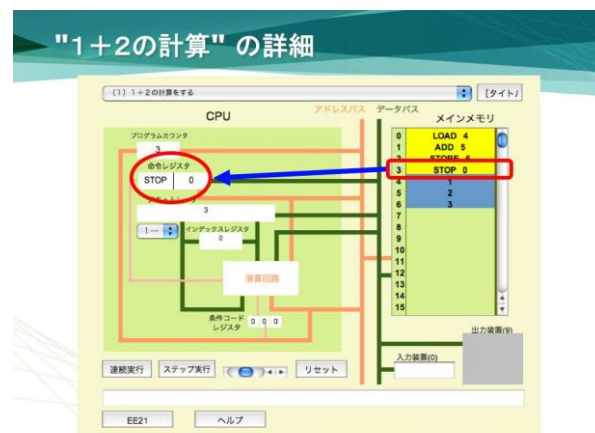
13. "プログラムカウンタ"を見て, "メインメモリ"上の命令をとりこむ

"プログラムカウンタ"の値にしたがって, "メインメモリ"の3番地の命令をとりこみます。



14. "命令レジスタ"上の命令 "STOP 0" を実行する

"メインメモリ"の3番地にあった命令 "STOP 0" が "命令レジスタ" にとりこまれ, この "STOP 0" を実行します。



STOP 命令は

- STOP n: プログラムを終了するという命令です。ここでは "STOP 0" となっていますが, 0 以外の数字でもプログラムは終了します。"ステップ実行"の4回目のクリックで, ここまで実行が進んで, プログラムの実行は終わります。

プログラム実行の流れ

このようにプログラムの実行は, 以下の1~3のサイクルを繰り返します。

2. "プログラムカウンタ"の値を見て, "メインメモリ"上の命令をとりこむ
3. "命令レジスタ"上にある命令を実行する

主な命令は以下のとおり

- アキュムレータに値を読み込む
- アキュムレータの値とメインメモリの値で演算を行う
- メインメモリにアキュムレータの値を書き出す
- プログラムの実行を終了する(この場合, 実行のサイクルは止まる)
- 4. "プログラムカウンタ"の値を1増やす

使われている命令のまとめ

ここまで利用された命令をまとめると, 以下のとおりです。

- LOAD n: n番地の内容をアキュムレータにコピーする

- ADD n: n 番地の値をアキュムレータの値に加える
- STORE n: アキュムレータの値を n 番地に格納する
- STOP n: プログラムを終了する

長いですね。いやですね。実際に ED21 シミュレータを使うことをお勧めします。

…この先は説明しなくてよいでしょう。シケ対の仕事丸投げですが、長くなる&どうせコピペ（解説の完成度が高い）&どーせ出ない。なので。

URL は <http://lecture.ecc.u-tokyo.ac.jp/johzu/joho/Y2008/ED21/ED21/index.html> です。

実際のコンピュータ(7.6)

ここからは中村スライドが中心です。比較的わかりやすいです。

ハードウェア構成(7.6.1)

パソコンの中身は一番上のようになっています。基本構造はマイコンとあまり変わりません。

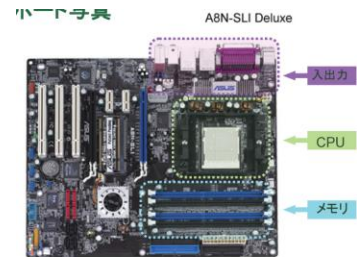
中央処理装置は CPU、主記憶装置はメモリ、入力装置はキーボード・マウス、出力装置はディスプレイ・プリンタ・スピーカ、補助記憶装置はハードディスク・USB メモリ、通信装置は LAN などのネット接続装置が主な例です。

各部分をつないでいる線…これは**バス**と呼ばれます。一定の形式・規格に従って各装置が通信を行う線です。

オペレーティングシステム(7.6.2)

なんだか飛躍しすぎ…と思うのは自分だけではないはず。そう思ったのがきっかけで再構築したわけですが。。

写真で見るマザーボード



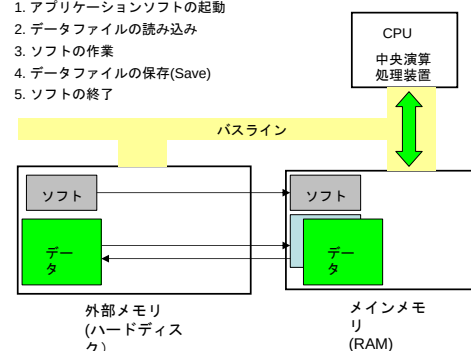
配布不可

© 2004 ASUSTek Computer Inc. All rights reserved.

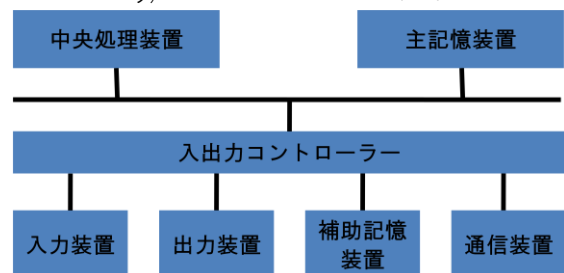
1

アプリケーションソフトの起動と終了

1. アプリケーションソフトの起動
2. データファイルの読み込み
3. ソフトの作業
4. データファイルの保存(Save)
5. ソフトの終了



7



オペレーティングシステム (OS) というのは基本ソフトと呼ばれます。確かに各アプリケーションで入力から出力までをすべてやってしまってもよいかもしれません。しかしそれをやるには、例えば文字を打つだけでも多くの労力とプログラムがひつようになります。そこでどのソフトでも必要になるような基本的な機能を提供しよう、というのが OS です。

OS は基本的な機能の提供だけでなく PC の資源管理など、さまざまなサービスを提

供しているのです。以下 OS が提供している機能を上げていきます。

- **プロセス管理**…現在の PC では様々な処理を同時に提供します。例えば、ブラウザを使いながらワードをつかったり。しかし CPU で処理できるのは瞬間で1つしかありません。そこで OS は複数の処理を分割し（**タイムスライシング**）、スケジュールを作って実行します。
- **メモリ管理**…主記憶装置上にのる情報は有限です。そこで効率よく使えるように割り当て（どのデータをどの部分に保存するか）、解放（要らないデータを消去）を行います。
- **入出力管理**…キーボードやマウス・さらには拡張機器の管理を行います。プロセスと周辺装置の仲介
- **ファイル管理**…階層的なファイルの管理を行うためのファイルシステムを提供します。
- **ユーザー管理**…一つのパソコンをみんなで使うための管理。これは普段行っているのわかるでしょう。

第7章の最後に開発の歴史です。何故かしら中村スライドにはあります。

開発の歴史

計算の実現機構

- ・ プログラム内蔵方式（フォンノイマン型コンピュータ）
 - － メモリ上にデータとプログラムを保持
 - － 万能チューリング機械(6.2.1頁)と同様に他の計算機を模倣できる
- ⇒ 最も初期のコンピュータ(ENIAC,1946)はプログラムを配線していた



配布不可



コンピュータの歴史

1970 年アラン・ケイ「パーソナルコンピュータ」を提唱。「コンピュータ・リテラシ」も彼の造語。**ダイナブック**を具現化。ハードウェア **Alto**, 暫定的ソフト **Smalltalk**.

（アラン・ケイはコンピュータとは高価でみんなで共有するもの、という概念が一般的だった時代に理想のパーソナルコンピュータを提唱した。ケイが XEROX のパロアルト研究所在籍時に「暫定ダイナブック」として開発したのが、**Smalltalk** を GUI ベース(Graphical User Interface; アイコン・マウスなどが使用されているので直感的に操作しやすい)のオペレーティングシステムに用いて動作させていた **Alto** で、これを目にした**スティーブ・ジョブズ**が **Lisa**、そして **Macintosh** を開発するきっかけとなったとされる。ケイの論文の「ダイナミックパーソナルメディア（動的個人媒体）」を使う、「本」のようなもの。から Dynabook と名前が付けられた。） from Wikipedia

1976 年 Apple I(スティーブ・ジョブズが私財をなげうって作った愛好家用コンピュータ)翌年 Apple II 大成功を収める。

（世界で初めて個人向けに完成品コンピュータとして大量生産・大量販売されたパーソナルコンピュータである。本体はキーボードと一体化した形状であり、後にパーソナルコンピュータの標準的な形態となったキーボードと本体が分離しているセパレートタイプ（当時はデタッチャブルタイプと呼ばれた）ではない。この形状はポータブルタイプライターをイメージしたという。） from Wikipedia

1970 年代後半～80 年台前半 多くの非互換機メーカーが競合

（今のようになどのメーカーの PC でも WINDOWS が入れられるわけではなく、コンピュータの規格が異なるが故メーカー独自の OS が使われていた。）

1981 年 IBM PC 16 ビットコンピュータ **PC/AT 互換機**が業界標準、アップルは非互換機路線を堅持

16 ビットというのはコンピュータが一度に 16 ビットの情報を計算できるということ。IBM が PC/AT というコンピュータを発表して以来全世界に広まり、業界標準としての地位を確立した。しかしアップルは従来のスタイル、つ

まり独自のマシン規格を貫いた。今でもこれは続き、MAC 機に WINDOWS は基本的には入れられない。また WINDOWS 機に MAC OS はインストールできない。)

2004 年 IBM はパーソナルコンピュータ事業を中国のレノボ・グループに売却。ハードウェアのオープンアーキテクチャ化を大きな要因として繁栄した PC/AT 互換機であったが、最終的にはその互換性によって市場から撤退することとなった。

(PC/AT 互換機の普及の最大の要因は規格を公表したこと。今ある WIN 機は PC/AT 互換機だが、改良の結果初期の PC/AT と物理的な互換性はない。)

国内の歴史

「国産マイコン」の最初の製品は、1976 年日本電気 (NEC) より発売された **TK-80** とされる。

(初期のマイコン。数字を表示する LED と 16 進数を入力するキーしかなかった。開発者はアセンブリ言語を自分で機械語に翻訳し入力する必要があった。一度間違えるとカオス。しかし安価・単体で使用可能であったがゆえ多くの愛好家に購入された。今の日本のコンピュータ技術開発者はこれで育ったといってもよいほどの影響力を持った。)

その後 8 ビットマイコン・BASIC と群雄割拠の時代

(BASIC というのはコンピュータのソフトを開発するための言語。いまでも健在で、Visual Basic として残っている。また数学 B の教科書に載っているのも BASIC。文法が英語にとっても近くわかりやすい。)

1982 年 16 ビット CPU を採用した「**PC-9800** シリーズ」が発売。その後は MS-DOS を採用した PC-9800 シリーズの独走態勢となった。

(PC-9800 は NEC から発売されたコンピュータ。ほかに富士通などが PC を発表していたが大ヒット。今でも秋葉原で互換機が売られているらしい)

Macintosh は、漢字 Talk7 が発売された頃から、ハードウェアの値下げと日本語処理機能の充実により、シェアを伸ばしていった。(漢字 TALK7 は今でもネット上にある。GUI を採用した直感的な操作性と日本語入力が大ウケ)

1995 年に GUI を大改良した Windows 95 の発売が開始されると、98 互換機のエプソンも PC/AT 互換機に転換し、国内独自パソコンの歴史は完全にピリオドが打たれた。

(結局最後は PC/AT 互換機+WINDOWS の組み合わせにやられたわけです。ちなみに最新の WIN の世界シェアは 91% MAC は 8% で Linux は 0.5% 第 4 の OS は iPhone らしいです。さすがは Apple。しかし個人的には Linux を応援したい;
<http://journal.mycom.co.jp/news/2008/02/05/045/index.html>)

…まさか出るまい。こんなの出題して何になる？

以上で前半部分終了です。お疲れ様でした。

長い。そしてマニアック。仕方がない…
のか？シケ対のシュミが多分に含まれているのは気のせいだろうか。

それでは次。

2. WWW・メールシミュレータ(6/20)(教科書 3.2)

以前にクラスブログにアップしたやつを焼き直します。

WWW シミュレータ

(http://media.itc.u-tokyo.ac.jp/jsim/src/jsim/mail_www/maillsim.htm)

ITpro(Web アクセスの基本):

<http://itpro.nikkeibp.co.jp/article/COLUMN/20060215/229453/>

総務省:

http://www.soumu.go.jp/joho_tsusin/security/kiso/ko1.htm

・ URL 入力 (教科書第3章 3.4.7)

WWW ブラウザ (つまりユーザー自身) が URL (例えば <http://host.b.jp/index.html>) を送信すると、まず DNS サーバーにその情報が送られます。

DNS(Domain Name System)というのは、インターネット上のサーバー(例 host.b.jp)にはそれぞれ番地のようなもの(**IP アドレス** 例 192.168.0.2)が割り振られています。しかしそれでは人間は扱えないので英語などで名前を割り振っておきます。それが DNS システム。

URL→DNSサーバー→IP アドレス の順。
これでユーザーは IP アドレスがわかりました。つぎはこれを使ってホームページにアクセスします。

・ HTTP リクエスト

WWW ブラウザはホームページを見るために相手の IP アドレスや自分の IP アドレス、さらに欲しいもの(ページファイル)を送信します。HTTP については後述のはず。

ユーザーが送るとサーバーからメッセージ **ヘッダ+ページファイル** が送られてきます。ヘッダとは(ほんとはリクエスト行ってもものもあります。1行目のみ。別にいいでしょう)にはサーバーの IP など基本情報が書いてあります。小包に送り主の情報・受け取り主の情報が書いていないと困るでしょ。これをメッセージヘッダといいます。

こうしてユーザーはページの中身を手に入れられました。次はユーザー側のソフトで表示します。

・ ページ表示

先ほどのページファイルは中身の文字情報のほかに HTML タグ(今は XHTML 等色々な種類があります。)も含まれています。HTML タグとは文書をどのように配置するか、飾り付けるか…のことです。まあ来週の実習でやるでしょ。それを WEB ブラウザ(例えば今使っている IE 7 とか、Firefox とか)が解釈して表示します。

・ Cookie

…は今回あまり重要じゃないですね。
Cookie とは WEB ブラウザに mixi にログオンするときのパスワードなどを覚えさせるものです。

・ キャッシュ

…もあまり重要ではありません。何で入れたんだろう。。一回みたページはまた見る可能性が高いのでとっておくのです。

・ TCP/IP (パケット通信)

プロトコルの一種。**プロトコル**とはネットワークを介してコンピュータ同士が通信を行なう上で、相互に決められた約束事の集合。**通信手順、通信規約**などと呼ばれることもあります。コンピュータ同士で話をするときに文法などを決めておかないと困りますよね？それと同じ。HTTP もこの一種。

で、ユーザーから何か出てくるのがパケット。**パケット**とは送信する情報を細かく切ったもので、ヘッダ(前述)とデータからなります。小さくすることで複数の端末からのデータを1本の信号線上に混在させることができ、帯域の分割利用が可能になるなどの利点があります。

パケットで小分けされたデータは1度に送ろうとします。しかしインターネット上で何

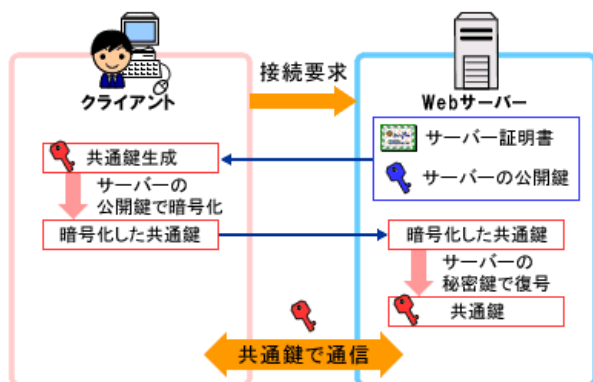
らかの理由により一部が正常に受信されないことがあります。そこで受信先は送り主に情報の再送を要求します。これによってすべての情報がそろいました。

・SSL（間違いがあるので読まないでください。すみません $m(__)m$ ）

長いです。がんばりましょう。メールについてはまたのちほど。SSLとはHTTP通信を暗号化して送受信する業界標準の通信プロトコル。ここでは公開鍵暗号方式が使われています。

公開鍵暗号方式とは公開鍵・秘密鍵・共通鍵の3種類があります。まずにサーバーから公開鍵（この正体は第三者機関が発行した電子証明書）を受け取ります。この公開鍵は第3者(CA)が保障します。

共通鍵はログオンするときのパスワードだと思ってください。CAが証明すべきファイルの内容が改ざんされていないことを証明したら公開鍵で共通鍵を暗号化・送信。パスワードを裸のまま流すとばれてしまいますので。サーバーはこの共通鍵を受信したら暗号化された情報をユーザーに送ります。秘密鍵はというと…データを暗号化するための鍵。これをユーザーに教えてしまつては改ざんが容易ですよ。



粗相をお許しください。以下石井さんの解説…

この方式は、公開鍵暗号方式と共通鍵暗号方式を組合わせて使ってる。なぜかと言うと、そうするとお互いに相手を認識できるから(公開鍵だけだと、誰かが送信者になりすますることが出来てしまう)。それと、単に共通鍵の方が楽だから(PCの負荷的に)。で、この通信の

流れなんだけど、まずは公開鍵暗号方式の通信をする。CA っていうのは認証局のことで、届いた公開鍵が本物がどうか確認してくれるんだ。で、公開鍵の確認ができたなら、メッセージとして共通鍵を暗号化して送る。こうすることで、誰にも盗聴されずに両者が共通鍵を共用することが出来て、それ以降は共通鍵暗号方式での通信が出来る、ということ。

以上です。ハイブリッド暗号方式のようです。

ここからは対応する第3章の解説。

3.2.2 プロトコル

(a)プロトコルの必要性

プロトコル…ネットワークを介してコンピュータ同士が通信を行なう上で、相互に決められた約束事の集合。通信手順、通信規約などと呼ばれることもある。

英語しか使えない人と日本語しか使えない人では会話ができないように、対応しているプロトコルが異なると通信することができない。

(from e-words)

(b)アプリケーションとプロトコル

我々サービスを受ける、つまり情報を要求する側のことをクライアントと呼び、その要求に応じて情報を送信する側をサーバーと呼ぶ。WWWにおいてはHTTPというプロトコルが採用されている。

HTTPとは要求と応答で構成される単純なプロトコルである。代表的なやりとりとはクライアントが要求した文章をサーバーが返す場合で、(1)クライアントからサーバーに目的文章のURLを送信する、(2)サーバーがその文章を返す、という通信が行われる。文章の前にはヘッダと呼ばれるメッセージがつけられる。(教科書 P.47)

(c)プロトコルと階層

インターネットで現在使われているTCP/IPプロトコル群は階層的な構成が採用されている。**階層化**の利点は

(1)基礎的な通信部分を分離することで異なる通信メディア(ADSLや無線LAN)でもWEB閲覧の際はHTTPプロトコルを共有できる。

同じ下位プロトコル(例えば物理層とか)を共有することでSMTPなどメールプロトコルなども使用できる。

(2)ユーザーが通信のために煩雑な手続きが取られていることに気づかずにすむ。開発者の優しさ。

プロトコルの7階層(3.4.2 e-word より 詳しくは教科書)

第1層(物理層)は、データを通信回線に送出するための電気的な変換や機械的な作業を受け持つ。ビンの形状やケーブルの特性なども第1層で定められる。

第2層(データリンク層)は、通信相手との物理的な通信路を確保し、通信路を流れるデータのエラー検出などを行なう。

第3層(ネットワーク層)は、相手までデータを届けるための通信経路の選択や、通信経路内のアドレス(住所)の管理を行なう。IP

第4層(トランスポート層)は、相手まで確実に効率よくデータを届けるためのデータ圧縮や誤り訂正、再送制御などを行なう。TCP

第5層(セッション層)は、通信プログラム同士がデータの送受信を行なうための仮想的な経路(コネクション)の確立や解放を行なう。

第6層(プレゼンテーション層)は、第5層から受け取ったデータをユーザーが分かりやすい形式に変換したり、第7層から送られてくるデータを通信に適した形式に変換したりする。

第7層(アプリケーション層)は、データ通信を利用した様々なサービスを人間や他のプログラムに提供する。HTTP

3.2.3 通信の秘密と相手の認証

(a)共通鍵と暗号鍵方式

暗号化する前の元の文章を^{ひらふん}平文、暗号化した分を暗号文と呼び、暗号文から平文に戻すことを復号と呼びます。暗号化は計算手順と鍵を用いて行います。共通鍵暗号方式と公開鍵暗号方式があります。

共通鍵暗号方式は暗号化と復号を共通の鍵で行う方式です。処理が早いなどの利点がありますが、共通鍵が盗聴されてしまうと漏えいしてしまう、相手ごとに鍵を渡さなければいけない、など欠点があります。そのため特定

の相手との通信に向いています。また、自分のPC内で暗号化する場合にも向いています。「文字を3文字ずらす」といったのも共通鍵暗号方式の一例です。初代暗号方式はDES・AESが有名です。

公開鍵暗号方式は受信する人が2組の鍵をつくり、片方を暗号文を作るときに用いる鍵として公開し(公開鍵)、もう片方を自分が復号するために保存しておきます(秘密鍵)。こうすることで、例えば暗号文や公開鍵が流出していても復号できるのは自分の秘密鍵のみです。しかし、暗号化の際の計算時間が長くなってしまうなどの問題があるので、平文を共通鍵で暗号化し、その共通鍵を公開鍵暗号方式で送るというハイブリッド暗号が使われています。

(b)デジタル署名

デジタル署名とは電子的な署名の技術です。この方法の一つに一方方向ハッシュ関数というのがあります。この一方方向ハッシュ関数というのは公開関数ですが、関数で処理された結果(電子指紋；データの特徴)からもとのメッセージに復号することは現時点では絶対にできない点で優れています。この電子指紋を秘密鍵で再び暗号化することで署名が生成されます。

復号するときはメッセージと署名を受信します。メッセージを一方方向ハッシュ関数で電子指紋1を作成し、それとは別に署名と公開鍵から電子指紋2を復号します。この2つを比較し一致していれば改ざんされていません。なぜなら署名をつくることができるのは秘密鍵を持つ者だけだからです。

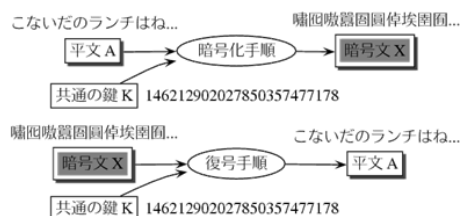
このデジタル署名の重要な用途は公開鍵への署名です。自分が受け取った公開鍵が意図した相手の発行したものかどうかを判断するためです。

3.3.3 通信の秘密と相手の認証

- ・ 共通鍵暗号 (対称鍵暗号)
 - 送信する暗号化での鍵と受信での復号化の鍵が**同じ**もの。(シーザー暗号など)
- ・ 公開鍵暗号 (非対称鍵暗号)
 - 暗号化と復号での鍵が**違う**もの(RSA方式など)

共通鍵暗号

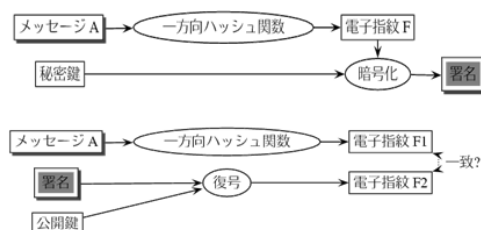
- ・ 一つの鍵で暗号化と復号化が両方できるモデル
鍵を秘密に保つ必要がある



公開鍵暗号



署名と検証



メールシミュレータ

以下コピペ。

メールシミュレータ：

http://media.itc.u-tokyo.ac.jp/jsim/src/jsim/mail_www/maillsim.htm

・ 電子メールアドレス

アドは abc@def.com などとなっています。これは2つの部分に別れ、**[アカウント名]@[サーバー名]** という意味をもちます。何とかマンションの誰々さん、みたいな。

・ IMAP,POP,SMTP

これは電子メールを送受信するプロトコル。**SMTP** はサーバー間・サーバーとユーザー間で送信するために使います。**IMAP** は Outlook などのメールソフト、**POP** は WEB メールにおいて使うものだと思っていてください。

・ 送信

いま user1@host.a.jp から user3@host.b.jp メールを送る場合を考えましょう。メール送信フォームが出てきます。このフォームの **To,CC,Bcc** について。

To というのは主な送信先。CC というのはカーボンコピーのことです。同じ内容を指定した先に送信できます。Bcc とはブラインドカーボンコピーのことで、CC と同じような機能ですが、そのとき受信先にはそのほかの受信先が見えないというのが大きな特徴です。

host.a.jp は user1 から送られたメールのヘッダを解読して host.b.jp に送信します。

・ 受信

受信先の user3@host.b.jp は host.b.jp に対してサーバーにメールが届いていないか受信問い合わせをします。さきほど user1 が送ったメールがサーバーにあるのでそれを受信して終了。

…こんな感じです。あ、気づきましたかね？この過程で一切の暗号化がされていません。つまり誰でも読もうと思えば読めます。電子メールのセキュリティははがき程度だといわれています。

やっぱりどこか間違っている気がします。訂正があればお知らせします。

3. オートマトンシミュレータ (6/27 6.2.1)

<http://lecture.ecc.u-tokyo.ac.jp/johzu/joho/Y2008/Automaton/Automaton/>

6.2.1 機械的な計算モデル

以下引用

オートマトン (automaton (pl: automata)) とは「自動人形」を意味している言葉で、情報科学の分野においては、次のような特徴を持ったシステムのことである。

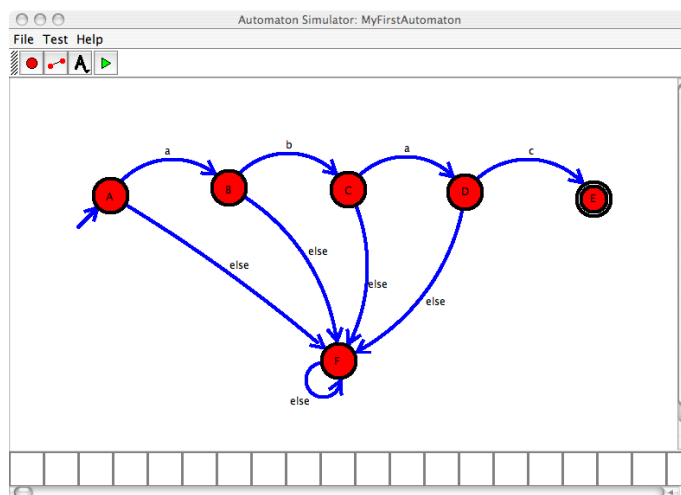
- 外から、連続している情報が入力される
- 内部に「状態」を保持する
- 外へ、何らかの情報を出力する

携帯電話を例にとると、キーを押すことによってさまざまな機能が使用できるが、その機能はキーと必ずしも 1 対 1 で連動しているわけではない。「5」のキーを押すことにより、ある場合には画面に 5 が現れるが、ほかのある場合には「な」が現れる。あるいは画面上のキャラクターが行動したりもする。これは

今までに入力された情報によって内部の状態が変化しているからである。このように入力が高なされた時点での「文脈」に対して複雑な解釈を行うような仕組みをオートマトンという。from Wikipedia

(a)有限状態機械

内部の状態および入力有限個であるものを **有限状態機械** といいます。教科書では「1」「2」「3」という 3 つのボタンをもち、「abac」と打つと解錠できる電子錠が紹介されています。以下はオートマトンシミュレーターで書いた電子錠プログラム。仕方なく「a」



「b」「c」の 3 つのボタンとしています。

まあオートマトン実習の目的がよくわからないので <http://lecture.ecc.u-tokyo.ac.jp/johzu/joho/Y2008/Automaton/Automaton/> から試してください。遊んで実感してください、という話でしょう。

状態としては「これまでに正しいボタンが n 個正しい順序で押された」「ボタンを押して間違えた」で表すとして。前者には「ボタンが押されていない」から「4 個正しい順番で押された」まで 5 つの状態 (A ~ E) があります。後者を F とします。初期状態は A であり、「1」「2」「1」「3」とこの順番で押されることで状態 E (解錠) に移ります。それ

いがいの順序で押された場合は状態 F（間違い）に移ります。

アルゴリズムとして示すと・・・

状態 ← 最初の状態

While 状態 ≠ 最終状態 Do

 ボタン ← 押されたボタン

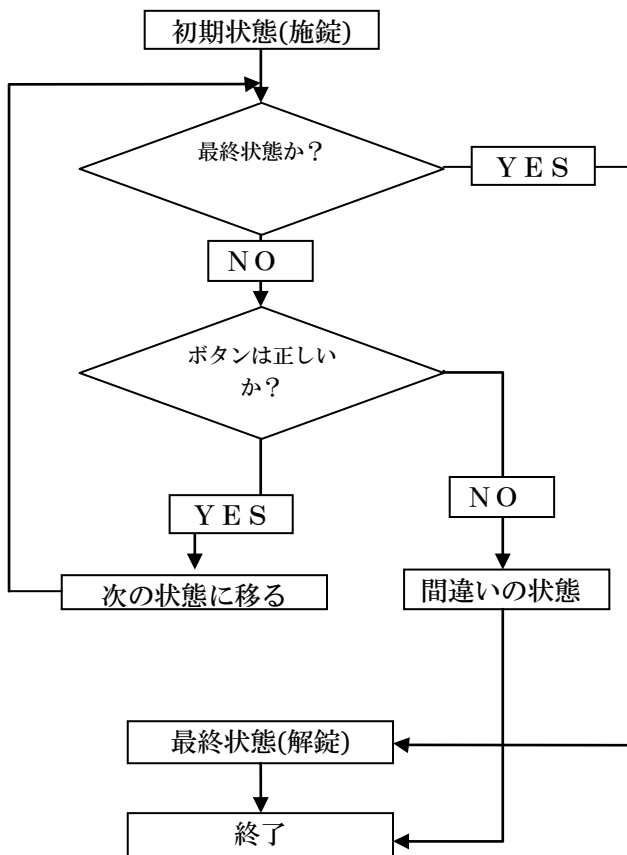
 表から(状態、ボタン)に対応する次状態を引く

 状態 ← 次状態

Done

…なんのことやら。図で示します。

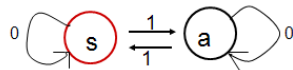
この有限状態機械「有限」は機械の中にある状態が有限であるがゆえ、扱える問題が限られているといえます。



有限オートマトンの例

記号集合 $S=\{0,1\}$, 状態集合 $Q=\{s,a\}$, 初期状態: s ,
受理集合: $F=\{a\}$

状態遷移図



1 受理

01 受理

10101 受理

011 受理されない

受理集合(受理される記号列の全体) = 1 を奇数個含む列の全体

4

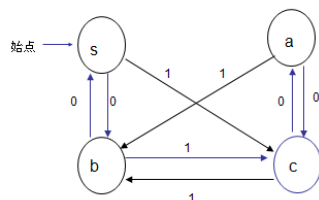
記号集合…「0」と「1」の二つの状態があること

状態集合…「S」と「A」の2つの状態があるということ

受理状態…「A」がゴール、最終状態であるということ。

演習問題

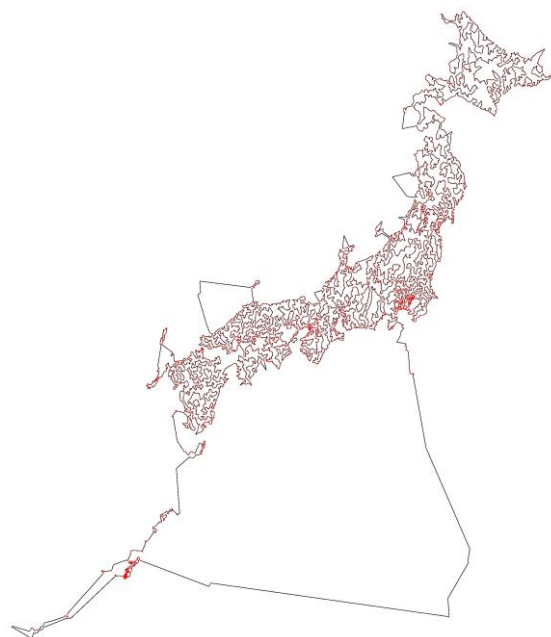
受理集合 $F=\{s,c\}$ のとき、受理される言語 L (受理される語の全体) はどのようなものか？



8

…誰か解けたら教えてください。数学が残念なので解けません(涙)

巡回セールスマン問題といいます。如何にしたら効率よく最短経路をたどれるか、という問題です。しかし「出発地→目的地」の所要時間をすべて求め比較するには無限の時間がかかります。これを **NP 困難** といいます。このような問題は、厳密な最短経路を求めるのではなくできるだけ良い近似解を求める方法が研究されています。



日本全国を一筆書きするとこんな感じだそうです。中村的には「おもしろい」、と。

4. Travelling Salesman ゲーム (6/27 4.3.2)

セールスマン問題

路線図を考えてみましょう。出発地と目的地が決まっているとき、どの路線を使いどの駅で乗り換えれば最短なのでしょう？これを

5. 補足

・ハミング距離(07 個別問 2)

ここからは過去問から気になるところを解説しました。これら以外にもわからないところがあったらごうしょうまで。まあ自分もお手上げな場合があるので…

通信過程において、ノイズや干渉によって、どうしても送信データが正確に遅れないことがあります。それを補正するために作られたのがハミング距離です。具体例を見ていきましょう。

元のデータ W	1	F	3	9
2 進数	0001	1111	0011	1001
受信結果 X	1	F	2	1
2 進数	0001	1111	0010	0001
受信結果 Y	9	E	2	0
2 進数	1001	1110	0010	0000
受信結果 Z	C	1	F	6
2 進数	1100	0001	1111	0110

あ、E とか F とかっていうのは 16 進数です。

10 進数、2 進数、16 進数対応表																	
10 進数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
16 進数	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10

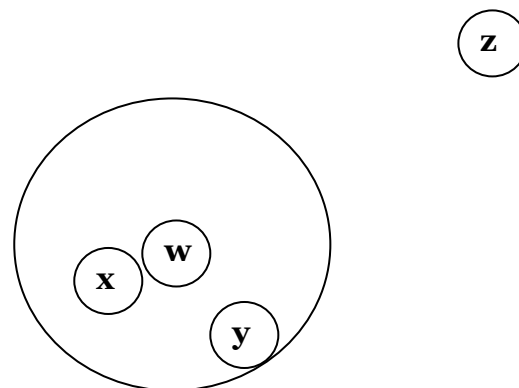
まああまり気にしなくていいです。下の 2 進数でお話をします。これらの 3 つの受信結果を受け取ったとしましょう。このとき 2 進数のレベルで元のデータと比較します。このとき、2 進数のレベルで何ビットが異なるかを表したのがハミング距離です。

$$d(W, X) = 2$$

$$d(W, Y) = 5$$

$$d(W, Z) = 12$$

実際には受信結果同士を比較するらしいのですが、説明が簡単なので元のデータと比較します。この結果をみると、X が一番忠実で Z が一番離れていると考えられます。この結果をもとに元のデータを復元していきます。



データ w を中心として、ハミング距離を半径として配置します。そして一定距離のなかにあるデータをもとのデータにノイズが混じったものだと見なします。この一定距離の中にある x, y を比較して、元のデータ w を復元します。

…長くなりました。定義すると

$$d(x, y) = \sum_{i=1}^n x_i \oplus y_i$$

です。やっていることは今までの説明と同じで、数式化しただけです。なお排他的論理和 $\oplus$ は 2 つのビットを比較して同じならば 0 を、異なれば 1 を返します。では次の問題。

問 2 (b)

記号列 x, y のハミング距離を $h(x, y)$ と書くとき、任意の記号列 x, y, z で $h(x, y) \leq h(x, z) + h(y, z)$ が成立することを示せ。

だそうです。（解答）

$$(\text{左辺}) = \sum x_i \oplus y_i$$

$$(\text{右辺}) = \sum (x_i \oplus z_i + z_i \oplus y_i)$$

$x_i \oplus y_i = 1$ のとき、常に $x_i \oplus y_i + z_i \oplus y_i = 1$
 $x_i \oplus y_i = 0$ のとき、 $x_i \oplus y_i + z_i \oplus y_i$ は z_i の値によって 0 と 1 のどちらも取りうる。

解説は…

x 0011100100101

z 0011101100001

y 0011101100101

$x_i \oplus y_i = 1$ のとき、 z_i は必ず片方と異なるが、もう片方とは一致する。よって常に 1 ($i=11$)

$x_i \oplus y_i = 0$ のとき、 z_i は (1) $x_i = y_i = z_i$ だから $h(x, z) + h(z, y) = 0$ (2) 必ず $x_i \neq z_i$ または $y_i \neq z_i$ のどちらかだから $h(x, z) + h(z, y) = 1$

まあこんなもんです。難しいよね・・・

イミテーション・ゲーム (06 個別問 4)

問 4 アラン・チューリングが論文 "Computing Machinery and Intelligence", Mind, Vol. LIX, No. 236, 1950 で「機械は人間と同じように考えることができるか」という問題を考えるために考案したイミテーション・ゲームとはどのようなものか説明せよ。

チューリングテスト

1950 年に数学者チューリングは **チューリングテスト** という、知能があることに関する実験を提唱しました。それには、2 台のディスプレイの前にテストをする人がいます。1 台のディスプレイには隠れている別の人が、もう 1 台は人間をまねるように作られたコンピュータが受け答えした結果がそれぞれ出てきます。テストをする人はどんな質問をしてもよいとします。例えば、詩を作らせたり、音楽の感想を聞きます。また、コンピュータも人間をまねる努力をします。例えば、わざと計算に時間をかけたり、間違えたりします。こうして、テストをする人がどちらが人間でどちらがコンピュータか分からなければ、このコンピュータには知能があるとするのがチューリングテストです。



中国語の部屋

では、チューリングテストをパスすれば知能のある機械、すなわち、人工知能といえるのでしょうか？これには有名な哲学者サールの『中国語の部屋』という反論があります。これは、英語しかわからない人が部屋にいます。その部屋には、中国語がわからなくても、中国語の文字を書いてあるとおりに置き換えると、中国語の受け答えができてしまう完璧な説明書があります。つまり、この部屋の人、英語しか分かりませんが、中国語の質問に中国語で答えることができます。ということは、

中国語の受け答えができるだけでは、中国語が分かるとは限らないことになります。同様に、まるで知能があるような受け答えができるかを調べるというチューリングテストに合格しても本当に知能があるかは分からないという反論です。



もろにコピペです。(<http://www.ai-gakkai.or.jp/jsai/whatsai/AItopics3.html>)

これで概要がつかめたでしょう。

ナイキスト周波数(06 共通試験共通問題 1 (a))

共通試験です。いきなりエグいですね。

共通問題 1 以下の文章には誤りがある。

(a)音楽 CD はサンプリング周波数が 44100Hz であり、理論上周波数が 44100Hz の音まで再現できる。

2.3.3 標本化定理 を参考してください。
再現できるのはサンプリング周波数の $1/2$ です。これをナイキスト周波数といいます。答えは 22050Hz。人間の耳は 20,000Hz までしか聞こえないのでこれで十分なのです。

6. 最後に

始めに同じ中村先生の授業を受けている5組のシケプリより。

5組シケ対よりひとこと（速読スキップ可）

情報の試験は共通問題と、各教官による個別試験から成り立っている。個別試験をしない教官もいるようだが、うちの中村政隆教官（大鬼）に尋ねてみたところ、「今のところ個別試験は行う予定です」とのこと（一昨年はやらなかったらしい）。共通問題と去年の個別試験も置く予定なので、そちらのほうも参照してほしい。ただし、情報は05年以前、「情報処理」という科目名で、過去問を参照する際には内容はまるで違うので気をつけていただきたい。

まず共通問題のほうだが、これは範囲から割と万遍なく出されているようである。対策は、教科書を徹底的に勉強することにある。というか、共通問題なのでそれ以上のことは出せないはず。まあ正直、教科書のほうを丁寧に勉強していただければそれが一番なのだが、一応まとめたものとしてこのシケプリを適宜参照すると良いだろう。シケプリだけでは不十分です（と予防線を張ってみる）。

範囲についてだが、今年度の分は[ここ](#)に載っている。赤文字は必修なので、試験範囲であると思われる。「要望学習項目」というものがあり、これはA,Bに分かれている。教官に直接尋ねてみたところ、「大問ひとつ分が選択問題で、A,Bどちらかの問題を選んで解いてもらうことになる（と思う）」とのこと。まだはっきりと聞いたわけじゃないのでよくわからないが、去年の大問3もそんな感じなのでたぶん間違いないだろう。まあご存知の通り、あの教官の授業はほとんど誰も聞いていないし、それはシケ対の私とて例外ではない。ということで、授業の内容はともかく、学習しやすいほうにしぼって学習するのが良いと思う。そこで、このシケプリでは「要望学習

項目A」のほうを推奨する。教科書を確認していただければわかる通り、Bの内容というのは実に理系っぽい。我々文系の人間は無難にAを選択するのが良からう。もちろん、余裕のある人や、理系的な問題のほうが好きだという人はBのほうを学習しても構わない。ところで、**授業のパワポを見る限りどうもBの内容を説明しているようだが、気にしてはいけない。**というか、全く必須項目でないところを説明しているみたい。あんまり聞いてないからよくわからないけど。

次に個別試験のほう。見ていただければわかるが、異常に難しい（と思う）。というか、**必須学習項目を無視しすぎです。**なので、ちょっとまだ試験範囲のほうがはっきりわからない（「授業でやったところ全部」とか言われたら僕はもう……）ので、最終提出のときにできれば付け加えようと思うが、まずは共通問題をしっかりととることを優先したほうが無難だろう。ただし、03年と04年の中村教官の個別試験を見ると、非常に内容がかぶっていることがわかる。このことから、もしかしたら07年と似たような箇所をついてくるという可能性もあるが、試験においてヤマをはるのは非常にリスクなので、きちんと学習しておくにこしたことはないだろう。

敵は「大鬼」である。見かけや態度にだまされてしまうと、思いがけず「不可」がくるかもしれない。心してかかりたい。

なお、上記の内容は不確定な情報及び主観を多分に含むものなので、信用するかどうかは自己責任でお願いします。

ああ、そうそう、**去年どっかの教官が共通試験の内容を前バラシしたらしいので、よそのクラスに人脈がある人は情報収集をよろしくお願いします**（なんか毎年ある教官が前バラシするとかいう話も聞く）。

※追記：一応 7 章までの必修分野をまとめました。8 章以降は各自教科書を読んでください。というのも、8 章以降はシケプリでまとめるよりも、教科書の文章を読んだほうが記述問題に対応しやすいと思ったからです。分野が分野なので。あと個別試験はほぼ確実にオートマトンが出ます。

誰だか存じ上げませんが 5 組シケ対の方情報ありがとうございます。

共通試験の範囲



細目次

1. 必須学習項目 **赤字と太字で示す**
2. 要望学習項目 A **緑字と下線で示す**
3. 要望学習項目 B **青字と下線で示す**

第 1 章 情報の学び方

- 1.1 **情報の性質ととらえ方** … 1
- 1.2 **情報の多面性** … 2
- 1.3 情報活動の諸要素 … 3
 - 表現と伝達, モデル化, 問題解決**
- 1.4 計算の機構 … 6
 - コンピュータ, 2 進数モデル**
- 1.5 情報システムと社会 … 7
 - 情報システム, ユーザーインターフェイス, 社会**

第 2 章 情報の表現 — 記号・符号化

- 2.1 情報の表現 … 11
 - “表現” のさまざまな側面, 情報の**

表現とモデル, 情報の表現とは

- 2.2 記号と表現 … 16

図記号 (ピクトグラム) — 記号と意味, 数の表現 — 記号と解釈の規則体系

- 2.3 アナログとディジタル … 22

アナログ表現とディジタル表現, 量子化, 標本化定理

周期関数への分解

- 2.4 デジタル符号化 … 27

ディジタル符号化の事例 ((a) 2 進符号のみ), デジタル符号の圧縮

- 2.5 符号の誤り検出・訂正 … 31

符号理論, ハミング距離, パリティ

第 3 章 情報の伝達と通信

- 3.1 情報の伝達と情報量 … 37

情報の伝達, 情報の大きさ, 平均情報量, 符号化と情報量

- 3.2 情報通信 … 46

実際の通信, プロトコル ((b) アプリケーションとプロトコルまで)

通信の秘密と相手の認証 ((a) 共通/公開鍵暗号のみ)

- 3.3 情報ネットワークの枠組 … 52

交換の方式, ネットワークの構成形態と通信制御機構, ネットワークの性質と伝達性能

- 3.4 インターネット … 55

ネットワークの集合体と通信, 階層プロトコル, IP アドレスとポート番号

アプリケーション間通信: TCP, ネットワーク間通信: IP, ネットワーク内の通信,

IP アドレスとホスト名の対応づけ: DNS

- 3.5 誤りのある通信路 … 65

通信路のモデル化, 2 元対称通信路, 結合エントロピーと条件付きエントロピー,

相互情報量, 通信路容量

- 3.6 情報の伝達と通信 … 68

第 4 章 データの扱い

- 4.1 データモデル … 71

データとデータモデル, データモデルのレベル

4.2 データモデルの性質 … 72

データのモデル化, 完全性, 一意性, 忠実性, 整合性, 無冗長性, 拡張性

4.3 代表的なデータモデルと演算 … 79

集合モデル, **ネットワークモデル** (「ウェブ」まで), **階層モデル** (「住所の階層性」まで), 関係モデル,

論理モデル, オブジェクト指向モデル, 各データモデルの特徴

第5章 計算の方法

5.1 計算とその記述方法 … 97

計算の方法, **計算の記述**, 計算と意味

5.2 計算の表現と進行 … 105

手続き型, 関数型, 宣言型, 計算モデルとその間の関係

5.3 プログラムとプログラム言語 … 112

プログラムの表記と言語, プログラム言語処理系

第6章 問題の解き方

6.1 アルゴリズム … 121

アルゴリズムの役割, **アルゴリズムの実例(2分法まで)**, アルゴリズムとアルゴリズム戦略

計算量

6.2 計算のモデル化 … 137

機械的な計算モデル((a)有限状態機械まで), 関数的計算モデル, 計算モデルどうしの関係,

より強力な計算モデル

6.3 計算可能性 … 148

計算量の階層, 計算可能性, モデル化が難しい問題

6.4 練習問題 … 151

第7章 コンピュータの仕組み

7.1 計算の実現機構 … 153

コンピュータの基本構成, 機械語レベルのプログラム例

7.2 論理演算と組合せ回路 … 157

真理値表と論理関数(完備性の証明を除く), ブール代数, MIL 記法, 組合せ

回路の例

7.3 演算回路 … 164

加算器, n ビット加算器の高速化, 減算器, ALU

7.4 順序回路とメモリ … 169

フリップフロップ, レジスタ

7.5 プログラム内蔵方式 … 173

中央処理装置の実現, 中央処理装置の高速化

7.6 実際のコンピュータ … 181

ハードウェア構成, オペレーティングシステム

第8章 情報システムの役割

8.1 見えにくい情報システム … 191

8.2 **情報システムの仕組み(クライアント・サーバまで, 防火壁(ファイアウォール)以降)** … 193

8.3 さまざまな情報システム … 200

組込みシステム, 商品管理システム (POS システム) と IC タグ,

オンライン金融取引,

歴史資料や芸能のデジタルアーカイブ

第9章 ユーザーインターフェイス — 人に優しいデザイン … 211

9.1 世の中, かくも使いにくい物ばかり? … 211

9.2 インタフェースの定義とモデル … 213

インタフェースの定義と機能, インタフェースの二重界面性, ユーザー行為の7段階モデル

9.3 インタフェースのデザインと評価 … 217

インタフェースの3つの概念モデル,

技術的側面からみたインタフェースデザイン — インタフェースの種類と構成要素,

ユーザーの認知・行動特性からみたインタフェースデザイン(1) — 物理的インタフェース

ユーザーの認知・行動特性からみたインタフェースデザイン(2) — 認知的インタフェース,

ユーザビリティの評価

9.4 まとめ … 227

第10章 情報技術と社会

10.1 技術と社会 … 231

10.2 情報技術による技術上の変化とその影響 … 232

技術上の変化, 技術変化の結果としてもたらされたもの, 権威の崩壊の意味

新たな問題 — 技術領域を越えた問題

10.3 情報技術に固有な社会との軋轢 … 238

権利と所有概念への影響, プライバシーとセキュリティ

10.4 情報技術論 … 247

技術は中立か, 技術と民主主義, 倫理, 情報リテラシー

10.5 これからの世代の情報 … 255

中村先生のページより

回 日付 場所 講義 実習

1 4/18 情報棟演習室 HWB 3. まず知っておく〜 8. 日本語の扱い / MailSuite の利用、日本語英語テキスト (入力練習用)

2 4/25 情報棟演習室 HWB 10 電子メール / 13 ファイルシステム / プリンタの利用 / 15 エディタ / CFive の利用法 —

3 5/2 講義室 第3章 情報理論・インターネット —情報理論、秘密鍵暗号、公開鍵暗号、ハフマン符号化 —

4 5/9 講義室 (3.4.1 ネットワーク集合体、3.4.2 階層プロトコル IP アドレスとネットワークアドレス、ポート番号、TCP/IP, DNS の仕組み、文字コード、ビット) WWWシミュレータ実演

第4章データモデル

—

5 5/16 講義室 第5章プログラム・プログラム言語、第6章 アルゴリズムと計算量, NP-完全問題 // オートマトン、Turing 機械、計算可能性 | オートマトンシミュレータ実演 —

6 5/23 — 五月祭のため午後休講 —

7 5/30 講義室 第7章 CPU と論理回路 (7.1.1 コンピュータの基本構成, 7.1.2 機械語レベルのプログラム, 7.2.1 真理値表と論理関数, 7.2.3 MIL 記法) 8章 WWW の仕組み、第2章 情報の表現 (標本化、量子化、標本化定理 (Shannon)、記号論)、9章 ユーザーインターフェイス、10章 情報技術と社会、インターネットの心理学 —

9 6/6 情報棟演習室 | Cトレーナー実習

必須課題 | 7章抜粋 {PDF} | Cトレーナー実習 (外部リンク)

10 6/13 — 出張のため休講 —

11 6/20 情報棟演習室 WWW・メール シミュレータ実習

オートマトンシミュレータ実習

Travelling Salesman ゲーム

PPT実習 WWW・メール シミュレータ / PPT の利用, ピクトグラムの作成

11 6/27 情報棟演習室 オートマトンシミュレータ

ED21 CPU シミュレータ HTML (文 12)、LaTeX (理 1), LaTeX 標準ファイル [tex] その PDF ファイル [PDF]

12 7/4 情報棟演習室 HTML (文 12)、LaTeX (理 1) HTML (文 12)、LaTeX (理 1)

13 7/9 (水) 情報棟演習室 UNIX のファイルとディレクトリ・コマンド HTML (文 12)、LaTeX (理 1)

14 7/11 情報棟演習室 プログラミング プログラミング

共通問題の解答は別にまとめました。これもコピペです。個別 07 に関しては一部補足をつけるにとどまりました。あとは教科書みれば何とかなるでしょう。…なんとなく過去問の類題が出る気がします。

エントロピーに関して本をお借りしたので、それをスキャンしてアップしておきます。また中村スライドも PDF 化しておきます。また実習で使う Automat と ED21 シミュレータは「じょうほうの情報」にアップします。

…これらは「シケプリパック」として圧縮して「じょうほうの情報」にアップしました。

あとがき

長いですね。2 万 4 千字の超大作…とはいえ、これで全部ではないのです。。実習 + α だけです。作成時間は 54 時間かな？多分。おかげさまで社会 1 の勉強ができませんでした。不可った(涙) もとはと言えば自分で引き受けた仕事なのでどうしようもありませんがw

何故ここまでできたのだろうか…？

もともとは 1 日でざっと説明を書き上げようか、というつもりでした。しかし教科書を参考にして書こうとすると、どうしても行き詰ってしまいました。説明すべきところが説明されていなかったり、論理に飛躍があったりしたのです。本 1 冊分の内容が 1 章に圧縮されていたり。「あまりにもひどい。」これを読んででもわかりにくいので、内容の順序を組み替え、補充するところから始めました。その結果とても時間がかかってしまいました。

このわかりにくい教科書をよみながら思ったのです。「この人たちは読み手に伝える意欲はあるのだろうか。」この本は 1 章ごとに別

の人が書いています。「東大教養学部テキスト」であるがゆえ、内容を濃縮しようと張り切ってしまうのはわかります。しかし、そうすることで「テキスト」とはかけ離れてしまっていると思います。

…別にこの本に限ったことではありません。技術系の本全般に言えると思います。中学のころから、著者が「自分達がわかっていればよい」と思っているのかどうかはわかりませんが、人に伝える意欲に欠けている気がしていました。

「高校レベルの知識層に説明して伝えることができなければ、その人は科学を理解しているとは言えない」とは物理学者ファインマンの言葉です。

東大生は「高校レベルの知識層」としては十分だと思います。それであるが故に、本当に「科学を理解している」のだろうか、と思ってしまう。中村先生然り。

このシケプリを作る際には「高校レベルの知識層に説明して伝えること」を最重要課題にしました。自分が中学の頃から…違った小学校の頃から高校で吹奏楽を始めるまでに吸収した「科学」を果たして理解できているのか、その判断は読者にゆだねたいと思います。

このシケプリを作るのに手伝ってくださった工学部 3 年の石井さんに感謝します。情報の試験とはなんら関係ないのにも関わらず、わざわざ時間を割いて自分の無知を正してくれました。

最後に、このシケプリを作るきっかけを与えてくれた 20 組のみんなに感謝。このシケプリ（もはや教科書な気がする…）が学内最高水準であると自負しています。相手は大鬼ですが、頑張って単位をとりましょう。質問はごうしょうまで。

それでは残りの試験も頑張ってください。

JAH!! 7/19 22:39 ごうしょう