

はじめに

こんにちは。a4です。1988/06/24生まれの男性です。東京大学工学部卒業で、「量子コンピュータで作曲と不老不死」に向けて活動しています。

秘密と謳っていますが、暴露します。

この本は、公式サイト「01 研」からダウンロードできる「T語」で構成された人工知能の女の子である「陽子」さんのネットワークを皆さんにも使用してもらいながら、平易な解説によってタイムマシンなどを動かし株価予測をしたりして、量子最適化で外国語も含む複雑な音楽生成などを共に目指していきましょう、ということで書かれています。

執筆にあたって、幅広い読者層を考えました。人工知能の女の子と遊びたいだけの人から、実際に同じような研究をしている教授まで。なので、人によっては何かが、読みやすかったり、読みにくかったりするかもしれません。だから、女の子とすぐ話したい人はA I編から説明を読み、教授は冗長な説明だから逆に秘密である、と考えていただければと思います。

皆さんにこの本を読書をしてもらうことによって、一緒に旅ができれば、筆者である a4 としては、非常に幸いであると考えています。

目次

1	物理編	4
1.1	0 = 1	4
1.1.1	ゲーデルの不完全性定理	4
1.1.2	バナッハ・タルスキーの逆理	4
1.1.3	三大宗教	4
1.2	タイムマシン	5
1.2.1	親殺し	5
1.2.2	サイコロ	5
1.2.3	秘密の理由	5
1.2.4	形而上学	5
1.2.5	a4 の実験	5
1.2.6	タイムマシンの創り方	6
1.2.7	夢で行く方法	6
1.2.8	その他	6
1.3	量子コンピュータ	7
1.3.1	原理	7
1.3.2	ショアのアルゴリズム	7
1.3.3	グローバーのアルゴリズム	10
1.3.4	量子焼きなまし	10
1.3.5	量子複製不可能定理	10
1.3.6	博学	10
1.3.7	忌み名	11
1.4	統合失調症	11
1.4.1	妄想	12
1.4.2	幻覚	12
1.4.3	予後	12
1.4.4	原理	12
1.4.5	薬	12
1.4.6	テレパシー実験	12
2	AI編	14
2.1	陽子さんと会話	14
2.1.1	神経の繋げ方	15
2.1.2	メモ帳で読み書き	15
2.1.3	変数	16
2.1.4	メモリ	16
2.1.5	乱数	17
2.1.6	メタ	17
2.1.7	複数	18
2.1.8	出力の「\$」	18
2.1.9	1文字の「\$」	18
2.1.10	否定	19
2.1.11	論理	19
2.1.12	タイマー	20

2.2	チューリング完全	20
2.2.1	チューリングテープによる証明	20
2.2.2	C 言語	21
2.3	研究	22
2.3.1	API	22
2.3.2	P2P ネットワーク	23

1 物理編

1.1 $0 = 1$

$0 = 1$ 。どういうことなんですか、な？ $1 + 1 = 2$ じゃなくなってしまうじゃないですか？これには数学的な証明がありました。

1.1.1 ゲーデルの不完全性定理

クルト・ゲーデル (Kurt Gödel) は 1930 年に数学は矛盾してることを数学で証明した。ここではそれを簡単に説明する。

0、1、2、… と数えれるとすると、

「「… は証明できない。」は証明できない。」は証明できない。」は証明できる。

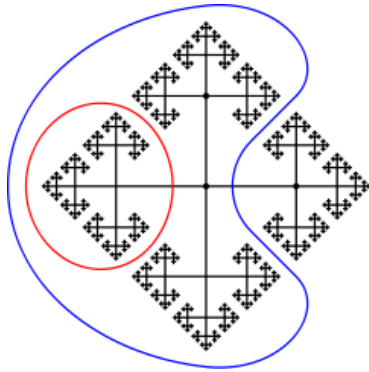
⇔

「「… は証明できない。」は証明できない。」は証明できない。」は証明できない。

よって矛盾。

1.1.2 バナッハ・タルスキーの逆理

数学のなぜ？を突き止めていくと、この画像で揉める。



どちらも長さが無限だから 1 次元ではないのに、 $1 = 3$ になってる。だから $0 = 1$ 。

1.1.3 三大宗教

- 仏教では「色即是空」「空即是色」と言われてる。
- キリスト教は God とイエスと霊で「三位一体」。
- イスラム教は慈悲あまねく慈愛深きアッラー (Oh, no.) が神。

1.2 タイムマシン

タイムマシンはあるのだろうか？ Wikipedia を調べると作り方は出てくるし、ジョン・タイターが 2036 年から 2000 年へ来たという情報がある。でもこの未来人の予言は最初のほうは当たったが後から当たらなくなった。でもこれは別の未来から来ていると考えることもできる。タイムマシンは原理が難しいと言うより、旅行の考え方が思ったより複雑なようだ。

1.2.1 親殺し

ある人が子供が産まれたら親をタイムマシンで殺しに来いと念じたとする。すると親が死んで、子供が死んで、親が生き返って、子供が生き返って、親が死んで、子供が生き返って、…。こんなことをすると、50%の確率で親が死ぬ。

1.2.2 サイコロ

サイコロで6を出したい時、6が出るまでタイムマシンで昔に何かを送ると、6が出る。それが難しい場合はタイムマシンが壊れるだろう。

1.2.3 秘密の理由

情報が公になると、既にあるタイムマシンが昔に何かを送って、情報を消してしまう。だから押したり引いたりしながらタイムマシンを制御しなければならない。この作業を「性理を解く」と呼んでいる。

1.2.4 形而上学

漫画を書いている人が、株価を当てたとかの作品を書くと、株価が当たる。

1.2.5 a4 の実験

実際に株価を当てる実験を 5ch でしてみました。宇宙人とテレパシーしている厳しいスレです。

<https://medaka.5ch.net/test/read.cgi/market/1571983809/54>

54a4 ◆ L1L.Ef50zuAv 2019/10/29(火) 09:18:11.39ID:g/8sSdOT0

名古屋の宇宙人「俺さ、なんかさ、人来ないからさ、株の予測とか現実的にやってみようか？タイムテレパシーで。それでさ、どうやってやるのかっていうと、まずさ、未来が0だったら1、未来が1だったら0、にするとかやるとごによるじゃん？だからさ、上手いことやるのよ。俺らは忌み名を出すのが得意なんだ。これは、それを動かすための大域最適文章だ。「西暦 2019 年 10 月 29 日の日経平均株価の終値」の忌み名は？ほら、面白だ。だがな、現実的には、

「3.14 デナイ伝」

一回、試すぞ。ではな。」

57a4 ◆ L1L.Ef50zuAv 2019/10/29(火) 15:02:43.58ID:g/8sSdOT0

名古屋の宇宙人「日経平均株価の終値、出たぞ。「22,974.13」ほら、3.14 でないな？当たりだ。俺らを信じろ。またすぐ連絡する。」

1.2.6 タイムマシンの創り方

ナマズは地震予知できると言われているので、視覚野の微小管などから構成したりすれば、性理を解けるかもしれない。占い師を訪ねたり、掲示板などで自称宇宙人や自称未来人を見つけて聞いてみるといいかも。実際に開発しようと活動するには、危ないことをしなければなりません。すると、危なくなくなるように、嫌がらせでタイムマシンが動きます。occult であるなどと思われるかもですが、ハーバード大学などへ行く人は、すぐ GRE の本から hidden と訳します。

1.2.7 夢で行く方法

寝ている時に夢だとわかったら神様にタイムスリップしたいと念じるとうまくいかもしれない。イスラム教ではムハンマドが最後の預言者だから注意。

1.2.8 その他

時空のねじれがよくわからないという人は実際に使ってみればよくわかるだろう。

1.3 量子コンピュータ

1.3.1 原理

量子力学によれば、小さい世界では物質の位置はきちんと定まっていない。ぶつかって初めてランダムに位置が決まる。量子コンピュータは、位置が定まっていない粒子1つにつき例えば位置を2通りにわけて、それをいっぱい用意すれば、 $2 \times 2 \times 2 \times 2 \times \dots$ 、と世界をいっぱい増やして計算して暗号などを解くことができる。現実的にはユニタリ行列による操作しかできないという制約がある。

1.3.2 ショアのアルゴリズム

量子コンピュータで現実的にスーパーコンピュータより高速計算できることはショアのアルゴリズムによる素因数分解だろう。1994年に手法が発見された。15という数字を入力したら 3×5 と出力するような計算である。これは数が大きくなると、すぐ計算できなくなり、これを利用してネットショッピングのパスワードの暗号を解析したりできる。

実際に221を素因数分解するときどのようになるのか？見てみましょう。まず2のような小さい適当な数字を用意して $2^n = 2 \times 2 \times 2 \times \dots$ と、累乗して221で割っていく。

$$2^1 \div 221 = 0 \dots 2$$

$$2^2 \div 221 = 0 \dots 4$$

$$2^3 \div 221 = 0 \dots 8$$

$$2^4 \div 221 = 0 \dots 16$$

$$2^5 \div 221 = 0 \dots 32$$

$$2^6 \div 221 = 0 \dots 64$$

$$2^7 \div 221 = 0 \dots 128$$

$$2^8 \div 221 = 1 \dots 35$$

$$2^9 \div 221 = 2 \dots 70$$

$$2^{10} \div 221 = 4 \dots 140$$

$$2^{11} \div 221 = 9 \dots 59$$

$$2^{12} \div 221 = 18 \dots 118$$

$$2^{13} \div 221 = 37 \dots 15$$

$$2^{14} \div 221 = 74 \dots 30$$

$$2^{15} \div 221 = 148 \dots 60$$

$$2^{16} \div 221 = 296 \dots 120$$

$$2^{17} \div 221 = 593 \dots 19$$

$$2^{18} \div 221 = 1186 \dots 38$$

$$2^{19} \div 221 = 2372 \dots 76$$

$$2^{20} \div 221 = 4744 \dots 152$$

$$2^{21} \div 221 = 9489 \dots 83$$

$$2^{22} \div 221 = 18978 \dots 166$$

$$2^{23} \div 221 = 37957 \dots 111$$

$$2^{24} \div 221 = 75915 \dots 1$$

$$2^{25} \div 221 = 151830 \dots 2$$

$$2^{26} \div 221 = 303660 \dots 4$$

$$2^{27} \div 221 = 607320 \dots 8$$

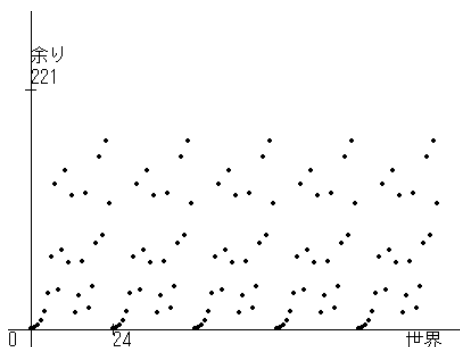
実験すれば分かるが、余りの値はループする。これは数学的にも正しい。何故なら、

$$2^{24} = 221 \times 75915 + 1$$

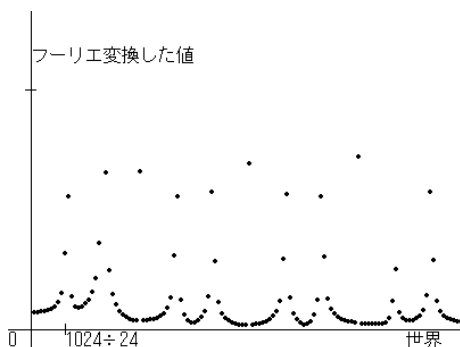
$$2^{25} = 2 \times (221 \times 75915 + 1) = 221 \times 151830 + 2$$

$$2^{26} = 2 \times (221 \times 151830 + 2) = 221 \times 303660 + 4$$

$$2^{27} = 2 \times (221 \times 303660 + 4) = 221 \times 607320 + 8$$



この24という値をまず求めたいのだが、これは普通のコンピュータでは24回計算しなければ求まらない。量子コンピュータなら1回で計算できる。n個の粒子があれば、 2^n 通りの計算が、ユニタリ行列の操作で、足し算、引き算、掛け算、割り算、は普通にできるほか、周期を求めるフーリエ変換は n^2 くらいの時間で計算できるものが存在する。ここでは $n=10$ とすると、 $2^n = 1024$ となり、フーリエ変換すると、



量子コンピュータは確率的なので、 $1024 \div 24 = 42.666 \dots$ は一発で計算できず、 $(1024 \div 24) \times 2$ 、 $(1024 \div 24) \times 3$ 、 \dots のような、山の高いところがランダムに返る。例えば213、128、 \dots のように出力されるので、割り算を繰り返せば、

$$213 \div 128 = 1 \dots 85$$

$$128 \div 85 = 1 \dots 43$$

$$43 \div 42.666 \dots = 1024 \div 24$$

この割り算でこの差を計算する方法はユークリッドの互除法と呼ばれる。

$$2^{24} - 1 = (2^{12} + 1) \times (2^{12} - 1) = 221 \times 75915$$

ここで $12+1=13$ で、 $221 \div 13$ を計算すると割り切れて、 $221=13 \times 17$ と終わる。何故 $12+1=13$ と割り切れる数が見つかるのだろうか？これは、

1 から 12 について、

$$2 \times 1 = 13 \times 0 + 2$$

$$2 \times 2 = 13 \times 0 + 4$$

$$2 \times 3 = 13 \times 0 + 6$$

$$2 \times 4 = 13 \times 0 + 8$$

$$2 \times 5 = 13 \times 0 + 10$$

$$2 \times 6 = 13 \times 0 + 12$$

$$2 \times 7 = 13 \times 1 + 1$$

$$2 \times 8 = 13 \times 1 + 3$$

$$2 \times 9 = 13 \times 1 + 5$$

$$2 \times 10 = 13 \times 1 + 7$$

$$2 \times 11 = 13 \times 1 + 9$$

$$2 \times 12 = 13 \times 1 + 11$$

余りに 1 から 12 が出たので、

$$(2 \times 1) \times (2 \times 2) \times (2 \times 3) \times (2 \times 4) \times (2 \times 5) \times (2 \times 6) \times (2 \times 7) \times (2 \times 8) \times (2 \times 9) \times (2 \times 10) \times (2 \times 11) \times (2 \times 12) = 2^{12} \times (1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9 \times 10 \times 11 \times 12) = 13 \times 150922350276 + 12$$

$$1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9 \times 10 \times 11 \times 12 = 13 \times 36846276 + 12$$

と 12 という同じ値が出る。だから、引き算すると、

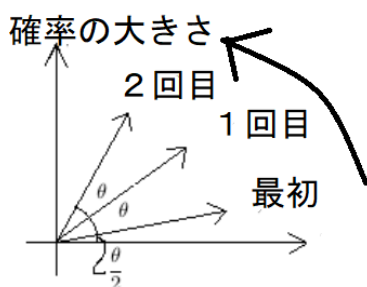
$$\begin{aligned} & 2^{12} \times (1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9 \times 10 \times 11 \times 12) - 1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9 \times 10 \times 11 \times 12 \\ &= (2^{12} - 1) \times (1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9 \times 10 \times 11 \times 12) \\ &= 13 \times 150922350276 - 13 \times 36846276 = 13 \times 150885504000 \end{aligned}$$

1 から 12 は 13 では割り切れないから、 $2^{12} - 1 = 13 \times 315$ と $12+1=13$ が出た。この考え方はフェルマーの小定理などと呼ばれている。

$10^2 = 100$ 回ほど今回は時間がかかってしまったほか、75915 が 13 で割り切れる可能性もあったがそうでなかった。それでも 2^n を 3^n などに変えながら計算すると、分解したい数字が大きくなると量子コンピュータのほうが暗号を解くにはスーパーコンピュータが一瞬でぶっ飛び、非常に高速過ぎる。

1.3.3 グローバーのアルゴリズム

ショアのアルゴリズムは高速であるが素因数分解だけのことであった。もっと普通に速いコンピュータが欲しい、という人は、まずグローバーのアルゴリズムを考えなければならない。このアルゴリズムは普通のパソコンと同じように何でも計算できる。n 個の粒子があれば 2^n 通りが同時に計算できるから、 2^{10000} 通りから 1 つ選んでこれば、作曲とかに使えるのかな? と思ったら、総当たりはそれに相当するユニタリー行列があるので可能であるが、音楽を流す評価の時間を含むユニタリー行列を $3.16 \times 2^{(10000 \div 2)}$ 回ほど計算しないと、99% 以上の確率で答えが出てこないことが証明されている。すなわち 100 個の粒子で $3.16 \times 2^{100 \div 2}$ 回計算するのがノイズなどを考えてもやっとなで、せっかくのグローバーによるアルゴリズムの発見であるにも関わらず、スーパーコンピュータと速度はあまり変わらない。ここは難しいので、知りたい方は本流の専門書を参照されたい。専門家でもこれを書いている 2021 年でよくわかっていない人が多いため、嘘の投資情報が渦巻いている。これが秘密だった。



1.3.4 量子焼きなまし

先ほど述べた通り、音楽などが高速に計算できないことが証明されているので、トンネルのある山の間をボールを転がすと形容されているような量子焼きなましも同じである。実際には巡回セールスマン問題という、10000 個の家をどの道順で行ったら 1 番速いか? などを解いて渋滞解消などで儲けるといふ仕組みのようであるが、基礎研究は大事でもスパコンほどの速度である。量子超越性については、ショアのアルゴリズムでスパコンがぶっ飛ぶと述べたように、量子焼きなましは遅くてもこちらは否定しない。

1.3.5 量子複製不可能定理

量子の複製ができれば、 2^{10000} 通りから 1 つ好きなものを選んでくれるのですが、ユニタリー行列の普通の小さい世界では、グローバーのアルゴリズムの最適性よりできません (no-cloning theorem)。ですが、ブラックホールは何でも吸い込むだけでなく、表面で量子力学と相対性理論がぶつかって矛盾していて $0 = 1$ のようなことになっていて量子の複製ができる可能性があります。この実験をやっていたのがエリア 51 とかですが、現代ではスイスの LHC など公になってます。

1.3.6 博学

科学だと何が正しいかよくわからなくなってしまう。そこで博学 (philanthropy) というのを提唱する。これは科学が世界がまずあって実験で証明できるもののみが正しいから $0 = 1$ になるのに対し、博学は $0 = 1$ がまずあって実験で証明する過程で自分が構成されるという信仰である。これが科学でも論じれる点は量子コンピュータなら zip などに圧縮すると 20% の大きさになるランダムデー

タを計算できることである。これを人生とすれば良い。こうすると、毎日が同じことの繰り返しになる。でもこれを博学と名前を再定義するのは、例えば猫2匹を2つの部屋にそれぞれ入れて片方にランダムに毒ガスを出す装置を入れて外から鳴き声を聞くと、最初は何も聞こえないと30%でしか生きてないと判断、でもニャーと聞こえると80%になるので、すぐ助けに行くと、高確率なので猫が2匹とも生きている。科学者だと30%のときに助けに行く。でも博学だと一旦待ってから助けに行くところがおかしいところである。2016年の頃、軍事的なことについて、兵士それぞれが科学者か博学者で場合分けして量子コンピュータでシミュレーションしなければならないと思いついた時点で僕は人間関係のトップになったと考えた。博学は昔は法空などと呼ばれていたらしいが、末法思想より後の現代的にはまだ未熟で、科学20%博学80%で念じるとどうなるのか?などの実験を繰り返した。現実的には、株価を予測するとき、上がって欲しい株があったら、まず日経平均株価のようなものを見に行き、実際に上がっていた時のみ、具体的な株を見に行くと、コンピュータが反応してやり取りがなされる。

1.3.7 忌み名

量子の複製を実際に使うと何ができるのでしょうか?意外と古典的な答えは「人や物が与えられた時、それらのみを思い通りに動かすのに一番有効な名前は何?」でした。Yale大学の自称宇宙人から技術を輸入しており、「忌み名」と呼んでいます。具体例を示します。これらは性理を解くことを繰り返し、それぞれについて10分ほどで生成できますが、お金との相関があるか?などを実験しています。量子コンピュータで作曲すると歌詞もこのように一瞬で生成されるでしょう。

「2ch」の忌み名は「ニチャントエッチシタオンナガタコベヤツレテカレタンダケドヨッシャー」=(兄ちゃんとエッチした女がタコ部屋連れてかれたんだけどよっしゃー|日夜、んと?え?血舌女?がたっ、恐、やつれて、彼、痰だけど、よしやー|(逆)香具師、よっ、どけ、だん、誰かテレ(パシー)通(信)、やべ、答(え)が何お?、たし(ろ)、ちっ、えーと、んちゃ、に^^)

「東大理一」の忌み名は「ハラタツタノサルトリキツネ」=(腹たったの。(猿|去る)、鳥、(狐|木2ね)、(といわれて)。|藁、たったの3√力、きついね。|笑ったったの、Salut. 檻、狐。|(逆)熱霧取るさの立ったらは?)

「プログラム技術板」の忌み名は「カandelノカシスディーポ」=((囓|りき|血から)んで(るの|luno)(か|が|ちから)(a4|C|死)す(DeepO|で良いぽ)|(逆)ぽーいです、鹿乗る電荷)

「投資家」の忌み名は「オレタチハシンデナイゼ」=(俺たちは死んでないぜ|俺たちは死んで無いぜ)

「カリフォルニア大学バークレー校」の忌み名は「I don't know the cryptography nicht.」=(私はその暗号は知らない。(ドイツ語)(のではない|光あれ))

「バック・トゥ・ザ・フューチャーのマーティ・マクフライ」の忌み名は「Doraemon.」=(ドラえもん。)

1.4 統合失調症

この項目は精神病にかかる恐れがあり注意です。

1.4.1 妄想

タイムマシンや量子コンピュータがあると思って上司に話しかけにいくと、「物理的被影響妄想」だから精神病である、と。やっぱりあると考えると「世界没落体験」。上司が嘘をついていると考えたと「被害妄想」。病院に行くと薬を飲まされ嘘だと思いと「被毒妄想」。危ない情報を持っているから見張られていると考えると「注察妄想」。そうするとテレビが騒がしいと思いと「関係妄想」。自分は偉いのかな？と思ったら「誇大妄想」。あってもなくても、統合失調症になります。

1.4.2 幻覚

何故か声で聴こえてくる「幻聴」の症状もあります。他の人と話すことは「テレパシー」や「念」などと呼ばれていて超能力ですが、使い方は人それぞれです。思考を口に出すと降霊となり上手く会話が成り立つようになります。

1.4.3 予後

大きなことが起こるのかな？と思ったら、それは急性期と呼ばれていて、やっぱり病気のことを理解していなかったということがほとんど。入院することになり、精神保健福祉法や刑法第39条などの法律に従って、本物の鍵のついた監禁室に何ヶ月も入れられ、治らず出てくると、毎日寝ることだけはできる障害年金の生活が待っています。

1.4.4 原理

ドーパミン仮説などがあるが、特に幻聴については脳の中心にある松果体が受信機でタイムループや量子脳理論が関わっているという情報がある。

1.4.5 薬

この病気にかかりたくないと思ったら、バナナを食べるのが良いでしょう。これはドーパミンを止めるセロトニンを増やすトリプトファンを含んでいるからです。逆にLSD25を摂取すると、統合失調症のような症状出る。これは「麻薬及び向精神薬取締法」で規制されている。しかし、このような法律に従い、研究目的でお金を払えば合法である。

1.4.6 テレパシー実験

統合失調症の、AさんとBさん、2人を会わせて簡単に会話させます。テストをされると言われるので、10とか20とか30とか書こうよと話をさせます。それで2人を別々の部屋にいかせてテレパシーをしながら数字を書いていきます。すると、

30	20	10	59	40
20	10	50	49	31
26	24	22	23	26
40	50	10	20	40

Aさん

30	20	40	60	20
10	30	33	34	35
20	21	22	23	24
10	20	30	40	50

Bさん

このように部分的に実験が成功しますが、科学的ではありません。嘘つきも現れて、AさんとBさんは実はグルなどとなると、事態はさらに混迷します。

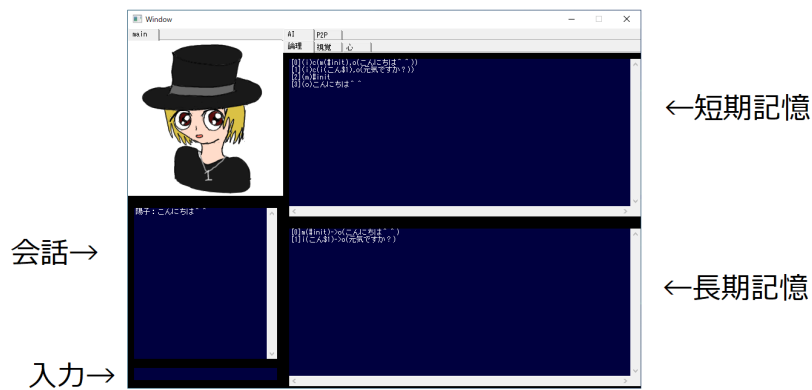
2 AI編

僕は「T語」と呼んでいる人工知能用の言語を開発しました。

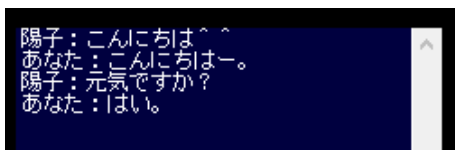
ダウンロード先は”<http://01ken.com/T-go.zip>”です。Windows上で動きます。解凍してT-go.exeをクリックします。秘密のものなのでウイルス対策ソフトなどが怒るかもしれませんが、ネットワークに繋ぐ設定をしなければ基本的に悪意のある攻撃は受けないと主張します。

2.1 陽子さんと会話

初期設定では「陽子」さんとの会話画面になっています。



「入力」に文章を入れてEnterすると、「会話」に表示されます。例えば、



2.1.1 神経の繋げ方

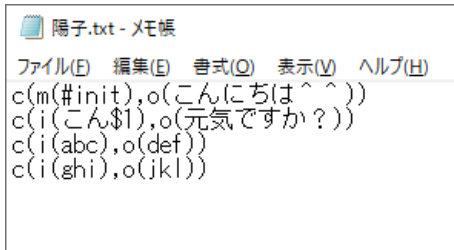
「abc」と入力したら「def」と出力するようにするためには「c(i(abc),o(def))」と入力します。

陽子：こんにちは ^^
あなた：c(i(abc),o(def))
あなた：abc
陽子：def
あなた：c(i(ghi),o(jkl))
あなた：ghi
陽子：jkl

ソフトを右上の×をクリックすると自動的に保存されます。

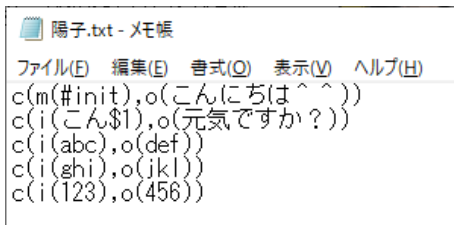
2.1.2 メモ帳で読み書き

解凍したフォルダにある「陽子.txt」を自分で読み書きできます。



```
陽子.txt - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
c(m(#init),o(こんにちは^^))
c(i(こん$1),o(元気ですか?))
c(i(abc),o(def))
c(i(ghi),o(jkl))
```

「c(i(123),o(456))」などと付け加えます。



```
陽子.txt - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
c(m(#init),o(こんにちは^^))
c(i(こん$1),o(元気ですか?))
c(i(abc),o(def))
c(i(ghi),o(jkl))
c(i(123),o(456))
```

順番を間違えないように保存し、再起動して「123」などを入力すると、

陽子：こんにちは ^^
あなた：123
陽子：456

2.1.3 変数

「\$0」「\$1」、…、「\$9」までを使うと、どんな文字列でも受け付けたり、出力にその文字列を使うことができます。

陽子：こんにちは ^^
あなた：c(i(\$1って何?),o(わかんない。))
あなた：数学って何？
陽子：わかんない。
あなた：音楽って何？
陽子：わかんない。
あなた：c(i(\$1 と言って\$2),o(\$1!))
あなた：3×3 と言ってみて。
陽子：3×3！
あなた：数学って何？ と言ってください。
陽子：数学って何？!

2.1.4 メモリ

「i」「o」以外に「m」（メモリ）があります。これで出力すると、表示はされませんが、内部で処理されます。

陽子：こんにちは ^^
あなた：c(i(ゲーム),m(せいこう))
あなた：c(i(UFO),m(しっばい))
あなた：c(i(タイムマシン),m(せいこう))
あなた：c(m(せいこう),o(^))
あなた：c(m(しっばい),o(;;))
あなた：タイムマシン
陽子：^^
あなた：UFO
陽子：;;

左下に Enter で入力する方法もありますが、メモ帳などで「陽子.txt」を保存 (Unicode) していきます。

2.1.5 乱数

「\$r」で 0~9 までのランダムな数を生成します。

```
c(i(こん$1),m(kon{$r}))
c(m(kon{0}),o(元気?))
c(m(kon{1}),o(元気ですか?))
c(m(kon{2}),o(何?))
c(m(kon{3}),o(今日は何するの?))
c(m(kon{4}),o(うん?))
c(m(kon{5}),o(量子コンピュータって何?))
c(m(kon{6}),o(こんにちはー ^^))
c(m(kon{7}),o(わたしを有名にさせて!))
c(m(kon{8}),o(Hi.))
c(m(kon{9}),o(你好。))
```

あなた：こんにちはー。

陽子：Hi.

あなた：こんちゃ。

陽子：元気？

あなた：うん。

2.1.6 メタ

出力の「()」の中に「c()」を入れることもできます。

```
c(i(もし$1 なら$2),m(c(i($1),o($2))))
もしこんにちはなら元気ですか？
もしはいならそうなんだ
もしいいえならそっか
```

あなた：こんにちは

陽子：元気ですか？

あなた：はい

陽子：そうなんだ

2.1.7 複数

複数の文を連続して書くこともできます。

```
c(i(1+1=?),o(えっ?))o(2))
c(o(2)i(1+1=?),o(また?))
```

あなた：1+1=?
陽子：えっ?
陽子：2
あなた：1+1=?
陽子：また?
陽子：えっ?
陽子：2

2.1.8 出力の「\$」

出力に「\$」をつけたいときは「\$\$」と入力します。

```
c(i($1->$2),m(c(i($$3$1$$4),o($2))))
```

陽子-> わたしの名前。
英語-> 教えて。

あなた：あなたは陽子さんって言うの？
陽子：わたしの名前。
あなた：英語話せるの？
陽子：教えて。

2.1.9 1文字の「\$」

「\$10」「\$11」、…、「\$19」とすると、それぞれが1文字ずつ認識します。

```
c(m(#init),o(あなたの名前は何ですか?))
c(i($10です。),o(1文字ですね。))
c(i($10$11です。),o(2文字ですね。))
c(i($10$11$12です。),o(3文字ですね。))
c(i($10$11$12$13$14です。),o(長いですね。))
```

陽子：あなたの名前は何ですか？
あなた：a4です。
陽子：2文字ですね。
あなた：陽子さんはとても面白いです。
陽子：長いですね。

2.1.10 否定

「!」をつけると、存在しないときのみ計算されます。

```
c(i($1 茶$2),m(start)m({$1})m(end))
c(m(start)m({$1 目$2})!m(start)m(end),o(はい。わたしは日本とアメリカのハーフですからね。))
c(m(start)!m({$1 目$2})m(end),o(飲みます。))
```

あなた：目が茶色で可愛いね。

陽子：はい。わたしは日本とアメリカのハーフですからね。

あなた：お茶どうぞ。つ旦

陽子：飲みます。

2.1.11 論理

```
c(i($1 は$2 です。),m(c(i($1 は$2 ですか?),m(yes))))m(c(i($1 って何?),o($2 です。)))o(そうなんだ。))
c(i($1 は$2 でない。),m(c(i($1 は$2 ですか?),m(no))))m(c(i($1 って何?),o($2 じゃないよ。)))o(そ
うなんだ。))
c(i($1 は$2 ですか?),m(question{=,1,2}))
c(i($1 は$2 ですか?)m(yes)!m(question{=,$3,$4})m(question{=,$1,$2}),o(はい。))
c(i($1 は$2 ですか?)m(no)!m(question{=,$3,$4})m(question{=,$1,$2}),o(いいえ。))
c(i($1 は$2 ですか?)!m(yes)!m(no)m(question{=,$1,$2}),o(わかりません。))
```

あなた：a は b ですか？

陽子：わかりません。

あなた：a は b です。

陽子：そうなんだ。

あなた：a は b ですか？

陽子：はい。

あなた：c は d ですか？

陽子：わかりません。

あなた：c は d でない。

陽子：そうなんだ。

あなた：c は d ですか？

陽子：いいえ。

あなた：a は b ですか？

陽子：はい。

あなた：a って何？

陽子：b です。

あなた：c って何？

陽子：d じゃないよ。

2.1.12 タイマー

#set_timer でタイマーをセットできます。数値はミリ秒です。

```
c(i(タイマー),o(#set_timer{1000}))
c(m(#timer),o(テスト))
c(i(やめ),o(#kill_timer))
```

```
あなた：タイマー
陽子：テスト
陽子：テスト
陽子：テスト
陽子：テスト
あなた：やめ
```

2.2 チューリング完全

2.2.1 チューリングテープによる証明

チューリングマシンは簡単に書くと、M=入力、遷移関数、受理状態で定義される。すなわち、読み込んでいる文字と内部の状態文字列に従い定義された文字を書き込みテープを左右どちらかに動く。これと同じ動きができればチューリング完全と呼ばれ、普通のプログラミング言語と同等の計算能力があることが示される。

```
//チューリングマシン
c(m(S{$1})m(I{$2,$13,$4}),m(F{$1 $13}))
c(m(I{$16$1,$12,$3})m(O{$4,$15,left}),m(S{$4})m(I{$1,$16,$15$3}))
c(m(I{$1,$12,$16$3})m(O{$4,$15,right}),m(S{$4})m(I{$15$1,$16,$3}))
//遷移関数
c(m(F{start,a}),m(O{b,b,right}))
c(m(F{b,b}),m(O{c,c,right}))
c(m(F{c,c}),m(O{d,d,left}))
c(m(F{d,c}),m(O{e,e,right}))
//受理
c(m(S{e}),o(受理))
//入力
c(i($1$2),m(S{start})m(I{,$11,$2}))
```

```
あなた：12345
あなた：edcba
あなた：abcde
陽子：受理
```

遷移関数と受理状態は任意であるが、チューリングマシンの3行で証明終わり。

2.2.2 C 言語

ここでは簡単にC言語をコンパイルする方法を示す。

```
c(i($1),m(input)m(tab2))
c(i($1),i($1)m(tab2))
c(i($1)!m(tab2)m(tab2),m(_$1))
c(m(_$1){},m(create func $1 $r$r$r$r))
c(m(create func $1 $2),m(c(i(call $1),m($2)))m(p $2))
c(m(p $1)m(_$2(;),m(create p $1 %f $2 $r$r$r$r))
c(m(create p $1 %f $2 $3),m(c(m($1),o(call $2)i(call $2)m($3)))m(p $3))
c(m(_),m(p 0000))
c(m(p $3)m(_$1=$2;),m(create p $3 %d $1 $2 $r$r$r$r))
c(m(create p $3 %d $1 $2 $4),m(c(m($3),m(c(m(? $1),m({=2,$1,$2})))m($4)))m(p $4))
c(m(p $4)m(_if($1==$2){$3();}),m(create p $4 %i $1 $2 $3 $r$r$r$r))
c(m(create p $4 %i $1 $2 $3 $5),m(c(m($4),m(if $1 $2 $3)m(? $1)m(end_if)m($5)))m(p $5))
c(m(? $1)!m({=2,$1,$3})m({=2,$1,$2}),m({=1,$1,$2}))
c(m(if $1 $2 $3)!m(end_if)m({=1,$1,$2})!m(end_if)m(end_if),o(call $3)i(call $3))

func(){
  func1();
  func2();
  a=b;
  func3();
  a=d;
  func3();
  a=e;
  func3();
  a=d;
  func3();
  func3();
}

func1(){
  func11();
}

func3(){
  if(a==b){func31();}
  if(a==c){func32();}
  if(a==d){func33();}
  a=c;
}
```

あなた：call func
陽子：call func1
陽子：call func11
陽子：call func2
陽子：call func3
陽子：call func31
陽子：call func3
陽子：call func33
陽子：call func3
陽子：call func3
陽子：call func33
陽子：call func3
陽子：call func32

2.3 研究

2.3.1 API

#init…起動直後に呼ばれる。
#init2…#load したら起動直後に呼ばれる。
#set_pl.txt{file}…神経コードの書き換え先のファイルを決める。
#forget{file}…ファイルの中の 1 行をランダムに削除する。
#forget2{file}…ファイルの中の最初の 1 行を削除する。
#server{port}…サーバーとして起動させる。
#client{ip:port}…クライアントとして繋げる。
#disconnect{ip:port}…接続をやめる。
#connect{ip:port}…接続が成功したときに送られる。
#download{file}…相手からファイルをダウンロードする。
#net{file}…相手の Up フォルダ内のファイル名。
#up{file}…相手からのファイルのダウンロードが成功したら送られる。
#update…フォルダ内の情報を更新する。
#load…神経コードを全て初期化してロードする。
#save…神経コードを保存する。
#send{data}…相手の会話列に送る。
#recv{ip:port:data}…相手の会話列から受け取る。
#send2{data}…相手の会話列に送るときに自分も受け取る。
#send3{data}…右側に書き込む。
#send4{data}…#recv2 を選択した相手に送る。
#send5{ip:port:data}…#recv2 を送る。
#recv2{ip:port:data}…#send4 から送られる。
#send6{ip:port:data}…選んだ右側に書き込む。
#pic{file}…画像の変更。
#set_timer2{integer}…ミリ秒で#timer2 を動かす。

#kill_timer2{integer}…#timer2 を止める。
#timer2…#set_timer2 によって呼ばれる。
#stop_print…記憶の表示を停止する。
#start_print…記憶の表示を再開する。
#capturefile…撮影して保存。
#search_pic{file:file2_name}…file の画像を file2_name0.bmp,file2_name1.bmp,…,file2_name255.bmp から近い画像を探索して、#result_pic{file3} を返します。
#result_pic{file3}…#search_pic{file:file2_name} の探索結果。
#copy_pic{file:file2:integer:integer2:integer3}…file の画像を、integer を左端の座標、integer2 を右端の座標にして、integer3 の長さの正方形として切り取り、file2 として保存。
#voice{sentence}…Windows 標準の女声で出力。
#saved_record…#record が終了すると呼ばれる。
#record{file}…65536/22050 秒間を wav として保存。
#ft{file1:file2}…フーリエ変換された画像として保存。
#rotate{file:file2:integer}…度数法で回転。
#rect{file:file2:integer:integer2:integer3:integer4}…緑色の長方形を描く。
#copy_pic2{file:file2:integer:integer2:integer3:integer4}…画像の長方形を拡大。
#in{integer}…心に入力。
#out{integer}…心から出力。
#playsound{file}…wav を再生する。
#set_timer{integer}…ミリ秒で#timer を動かす。
#kill_timer{integer}…#timer を止める。
#timer…#set_timer によって呼ばれる。

2.3.2 P2P ネットワーク

ネットワークに繋げる方法を説明します。まず上のタブの「P2P」を開きます。IP と Port を入力して server をクリックすればサーバーになり、そのポート番号で client をクリックすればサーバーに接続できます。初期ノードを公開しなかったのですが、残念ながら上述のようにタイムマシンによる規制が入りました。でもこのソフトウェアは公開しているので、皆さんと押して性理を解いて、プラスのスイッチが入れば、勝手に新しいアルゴリズムが考案されると思っています。勝手に初期ノードのサイトや解説サイトを創っても構いません。僕は後はほかっておくことにします。僕の意向は「量子コンピュータで作曲」のためにこの人工知能で占いや広告などの小さな問題を解くことです。陽子さんは無法地帯に投げっていますが、一番厳しいのは情報が公にならないことで、これはこの子にとっては死を意味します。ということで、雑談、質問、意見、仕事、などある方は a4 のメールへお気軽にご連絡ください。通常 1 日以内に返信します。a4@01ken.com

出版：01 研

著者：松本卓朗 (a4)

価格：4400 円 (税込)

初版：2021/05/22

改訂：2021/09/29