

目次

第 0 章	まえがき		
第 1 章	λカ娘探索ファイナル	@ark_golgo	1
1.1	プロローグ		1
1.2	AlphaGo 概要		3
1.3	囲碁のルールおさらい		3
1.4	AND-OR 探索		5
1.5	Min-Max 探索		9
1.6	α - β 探索		11
1.7	α - β 探索が囲碁ではうまく機能しなかった理由		11
1.8	原始モンテカルロ法		13
1.9	モンテカルロ木探索		13
1.10	AlphaGo 登場前の囲碁ソフト		14
1.11	Deep Learning		17
1.12	AlphaGo		17
1.13	今後の囲碁ソフト		19
1.14	参考文献		20
第 2 章	変態する昆虫の観察	@master_q	21
2.1	今日の宿題		21
2.2	観察		21
2.3	静的な意味とは何か		23
2.4	捕まえよう		26
2.5	変わらないと信じること		30
2.6	ライセンス		30
2.7	参考文献		31
第 3 章	IST(Internal Set Theory) 入門 (前編)	@dif_engine	33
3.1	プロローグ		33
3.2	解析学からの無限小概念の追放		34
3.3	述語論理		38
3.4	公理的集合論 ZFC		46
	会員名簿じゃなイカ?		58

第0章

まえがき

関数型イカ娘とは!?

Q. 関数型イカ娘って何ですか?

A. いい質問ですね!

Q. 九冊目なんて、こんなの絶対おかしいよ!

B. 僕達の役目はね。関数型言語とイカ娘の設計、実装、原理、応用に関する妄想を抜き取って、文章に変えることなのさ

関数型イカ娘とは、「イカ娘ちゃんは2本の手と10本の触手で人間どもの6倍の速度でコーディングが可能な超絶関数型プログラマー。型ありから型なしまでこよなく愛するが特に Scheme がお気に入り。」という妄想設定でゲソ。それ以上のことは特にないでゲソ。

この本は、コミックマーケット80での「簡約! λカ娘」、コミックマーケット81での「簡約!? λカ娘(二期)」、さらにコミックマーケット82での「簡約!? λカ娘(算)」に続く、さらにさらにコミックマーケット83での「簡約!? λカ娘4」、に続く、もーさらにさらにコミックマーケット84での「簡約!? λカ娘 Go!」、に続く、そしてコミックマーケット85での「簡約!? λカ娘 Rock!」、コミックマーケット86での「簡約 λカ娘 巻の七」、コミックマーケット88での「簡約!? λカ娘8」、に続く、九冊目の関数型イカ娘の本でゲソ。コミック連載終了に安部真弘氏に「おつかれさまでした!」の気持ちをこめて、関数型言語で地上を侵略しなイカ!

この本の構成について

この本は関数型とイカ娘のファンブックでゲソ。各著者が好きなことを書いた感じなので各章は独立して読めるでゲソ。以前の「λカ娘」本がないと分からないこともないでゲソ。

第1章

入カ娘探索ファイナル

— @ark_golgo

Contents

1.1	プロローグ	1
1.2	AlphaGo 概要	3
1.3	囲碁のルールおさらい	3
1.4	AND-OR 探索	5
1.5	Min-Max 探索	9
1.6	α - β 探索	11
1.7	α - β 探索が囲碁ではうまく機能しなかった理由	11
1.8	原始モンテカルロ法	13
1.9	モンテカルロ木探索	13
1.10	AlphaGo 登場前の囲碁ソフト	14
1.11	Deep Learning	17
1.12	AlphaGo	17
1.13	今後の囲碁ソフト	19
1.14	参考文献	20

1.1 プロローグ

「1勝4敗か…。内容を見る限り、『よく1勝した』と言うべきなのじゃろうが…。」

「そうでゲソね…。まさかこれ程とは思わなかったでゲソ…。」

大きなニュースになったので、読者の皆様もおそらく何のことかご存知であろう。そう、

「イ・セドル対 AlphaGo 戦」

のことである。入カ娘6で述べた通り、大半の研究者が、

「囲碁で AI が人間トップを超えるにはまだまだかかる。」

と思っていた。しかし、2016年3月、人間のプレイヤーとして世界トップクラスのイ・セドルが、Google が作った囲碁 AI の AlphaGo と5番勝負をして、わずかに1勝しか出来なかった。研究者の予想が見事に裏切られた訳である。

「しかし、お主は確か以前、モンテカルロ囲碁ソフトの強さは頭打ちになっていて、それを打ち破るには『攻め合い』や『死活』などの細かい一本道を読み切る能力が必要だと言っておったな。それをモンテカルロ囲碁に組み込むことはとても難しいことで、誰も成功していないとも。この AlphaGo を作った DeepMind の人達は、この短期間にそれを成し遂げてしまったのかね？」

「私は AlphaGo の論文を読んだでゲソが、そういう訳ではないでゲソ。爺さんも対局の内容を見て気づいたんじゃないか？ 実は、AlphaGo は、論文を読む限りは読みの力は大きかったことがないと思われるでゲソ。どうやら、大局観だけでイ・セドルに圧勝してしまったらしいんでゲソ。」

「確かに、対局の内容を見て何となく分かってはおったが…。ただ、すまんが僕は AlphaGo の論文を読んで理解するだけの英語力も AI に対する知見も持ち合わせておらん。よければ、AlphaGo とはいかなる物なのか、僕に説明してくれんかね？」

「勿論、喜んで説明するでゲソ。ただ、AlphaGo の凄さを理解するためには、囲碁 AI の歴史を知らなければならぬでゲソ。それに、んカ娘 6 を読んでいない読者もいると思われるでゲソ。さらに言えば、囲碁のルールを知らない読者も居ると思われるでゲソ。だから、長くなるでゲソが、まず AlphaGo の概要を述べた後、囲碁のルールとゲーム AI の基礎及び囲碁 AI の歴史も説明して、改めて AlphaGo の詳細を述べるでゲソ。長くなるでゲソが、付き合ってくれなイカ？」

「勿論構わんよ。茶と菓子を用意したから、休みながらで構わん。ゆっくり聞かせてくれ。」

さて、AlphaGo とはいかなる物なのか、んカ娘にじっくり説明してもらおう。

第2章

変態する昆虫の観察

— @master_q

2.1 今日の宿題

「ぱっと見たただけだと昆虫は不規則な動きをしているだけのようで、よーく観察すると規則が見つかることがあるじゃないか。」

ウチのクラスの担任はいつも授業にノリノリだ。

「このように一見動的に見えるモノでも、変わらない規則があることがあり、これを静的な意味と呼ぶことがあるでゲソ。」

変わらないこと... そんなものがあるのだろうか。全ての生物は死んでしまふし、統一理論は一向に定まる気配はない。これまで人類が組み上げてきた数学だって、根本的な矛盾が発見されたら、その全体を修正するしかないんじゃないか？

「そういうわけで、今日の宿題は昆虫の観察でゲソ！来週は皆に昆虫に見る静的な意味を発表してもらおうじゃないか。」

今は 21 世紀。火星への移住が計画されるような現代において、昆虫を観察することにはたして意義があるのだろうか... ヤレヤレだ。

2.2 観察

昆虫といえば以前とても小さな OS、**ChibiOS/RT**^{*1} を見つけた。この OS はバイナリサイズが小さく、コンテキストスイッチにかかる時間が短いことが特徴^{*2} で、8-bit MCU である AVR や 32-bit MCU である ARM Cortex-M シリーズ上で動く。また良くドキュメントも整備されていて好感がもてる。僕はこの OS を日々の工作にも使っている。

ところが、この ChibiOS/RT 上で動くアプリケーションを作るのは簡単ではない。その理由はいくつもあるが、その 1 つはシステム API の呼び出しマナーがわかりにくいことだ。ChibiOS/RT のような OS をリアルタイム OS と呼ぶが、このリアルタイム OS に習熟していれば、ChibiOS/RT に入門するのはたやすいかもしれない。しかし、デスクトップ OS 上でのアプリケーションだけの知識だと以下のような概念に馴染みがないことがあるのだ。

- POSIX スレッドではない独自のスレッド機構
- システムグローバルのロック
- システム状態
- 再入可能
- 割り込みの許可/無効

^{*1} <http://www.chibios.org/>

^{*2} <http://wiki.chibios.org/dokuwiki/doku.php?id=chibios:metrics>

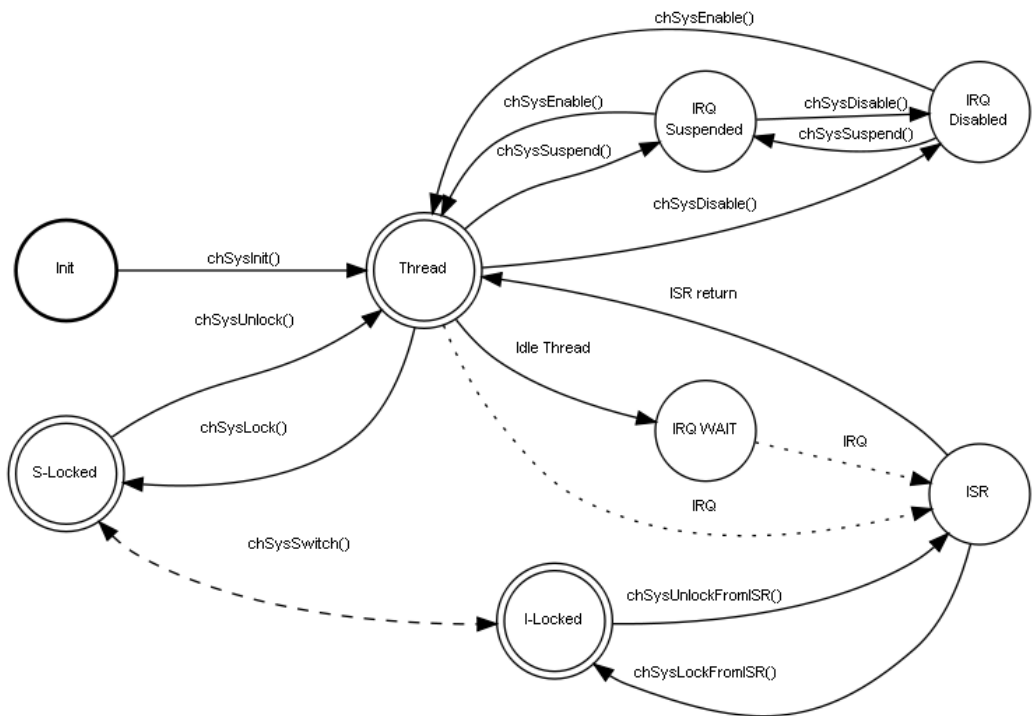


図 2.1: ChibiOS/RT のシステム状態

ChibiOS/RT の場合、これらの概念は具体的にどのようなになるだろうか。まずこの OS には図 2.1^{*3} のようなシステム状態がある。ChibiOS/RT では `main()` 関数は `Init` 状態で開始される。`main` 関数の中で自発的に OS の初期化関数 `chSysInit()` を呼び出すことで、この OS 全体は `Thread` 状態に遷移する。この状態においてシステムグローバルのロックを獲得する関数 `chSysLock()` を呼び出すと、システム状態は `S-Locked` に遷移する。ロック排他すべき処理が終わったら `chSysUnlock()` 関数でロックを解放するとシステム状態は `Thread` に戻る。

もし `Thread` 状態の最中にハードウェア割り込み (IRQ) が入るとどうなるだろうか？ この場合、実行中のプログラムは中断し、強制的に割り込みハンドラが実行される。そしてシステム状態は `ISR` 状態に遷移する。つまり割り込みハンドラは常に `ISR` 状態から開始されるのだ。割り込みハンドラの処理が終わって関数から返るとハードウェア割り込みは終了し、`Thread` 状態に戻る。そして同時に中断されていたプログラムが再開されることになる。

比較のために、POSIX 上でのプログラミングを思い出してみよう。システム状態と呼べるようなものはメインコンテキストとシグナルコンテキストの二種類しかなかった。シグナルコンテキストから呼び出せる関数、つまり再入可能な関数は POSIX で定義されているが、実装によりけりなのでシグナルコンテキストであまり凝った処理をすることは稀だろう。システムグローバルのロックもない。考慮しなければならないことは少なかつたはずだ。ChibiOS/RT のようなシステム状態は POSIX を実現している OS が吸収してくれているのだ。

ところが、ChibiOS/RT 上のアプリケーションは上記のシステム状態を把握しながらプログラミングする必要があるのだ。しかも各システム状態から呼び出せるシステム API は限定されていて、間

^{*3} http://www.chibios.org/dokuwiki/doku.php?id=chibios:book:kernel#system_states から引用

第3章

IST(Internal Set Theory)入門(前編)

— @dif_engine

目標と概要

超準解析 (nonstandard analysis) を行うための理論的な枠組みとしてエドワード・ネルソンにより創始された IST(internal set theory, 内的集合論) について解説します。ネルソンの論文は丁寧に書かれています。読者が述語論理や公理的集合論にある程度慣れていないと読むのは難しいと思われます。この記事の目標は、そのあたりの予備知識を補いつつ IST の入り口まで読者を案内することです。

3.1 プロローグ

夏が苦手だ。夏の草木や空気——そしてなによりあの太陽——それらが、まるで横溢するその生命力でわたしを圧殺しようとしているかのように感じてしまうのだ。まあこんなのは、うっすらとした霧につつまれて夏でもひんやりとしたこの紅魔館の、さらに地下の図書館にこもっていたいだけのわたしの言い訳なのかもしれないが。

いつものように遅く起きたわたしは、居室代わりにしている紅魔館の図書館の扉を開けた。漂ってくる紅茶の香り——この部屋に先客だなんて珍しい。

すました顔で茶を喫していたのはアリスだった。アリス・マーガトロイド。「人形遣い」の能力を持つ魔法使いだ。彼女の傍らには美しい人形があった。図書館のテーブルのその上にミニチュアテーブルが置かれており、その上に可愛らしいティーセットが並んでいた。このミニチュアティーセットからも湯気がうっすらと立ち上っている。ミニチュアテーブルの傍らの椅子に座る、ちんまりとした陶製の顔をもつその人形は服から顔にいたるまでアリスに似せて造られていた。

それにしても、とわたしは思う——この人形は、なんとアリスに似ていることだろう——と。それとも話は全く逆で、アリスがあの人形に似ているのかしら——などと埒もないことを考えてしまう。アリスがカップと皿をテーブルに戻すと同時に、人形が動き出す。人形は、その創り主を真似たような優雅な所作でティーカップと皿をとりあげ、うっとりとした表情で紅茶の芳香を味わっていた。

じっとその様子を眺めていると人形がついと顎を上げ、眼が合ってしまった。光彩をその裡にけがらせた宝石のような瞳は夢を見るようにはかなげで、わたしはたちまち魅入られてしまう。

「——挨拶もなしに人の顔をじろじろ見るなんて、礼儀を知らないのかしら」

声の主はアリスだった。

「あらごめんなさい——」

そう言いつつわたしは戸惑う。いまわたしは彼女の人形の顔を見ていたはずなのでは？ いやいや、それも気になるところだが紅魔館の食客とはいえわたしはこの部屋の主なのだ、連絡もなく訪れて

我が物顔でお茶を飲んでいる彼女のほうがよほど礼儀に反しているということにならないだろうか？

そんな疑問も、彼女の白皙の美貌に軽く跳ね返されてしまいそうで思わずわたしは黙ってしまったのだが――

まさにこの瞬間、わたしはある異変に気づいた――ミニチュアテーブルの周りの人形がいつの間にか二体が増えていたのだ。新しい人形の顔はこちらからは陰になって見えないが、その衣装には見覚えがあった――わたしの服だ。ならば、この新しい人形の顔がどんな様子かは予想がつく――きっとわたしそっくりに違いない。

アリスの後ろに回りこんで新しい人形の顔を覗き込みたいという気持ちを抑えながらわたしは考えた；一体いつアリスは新しい人形――おそらくわたしそっくりの顔をした――を取り出したのだろうか？

いっそアリスに直接聞いて見ればよかったのかもしれない。だが――なんと聞けばよかったのだろうか。「あら、素敵なお人形さんね、さっきまで一体しかなかったお人形が二体が増えてるわね。それともわたしの見間違いかしら？ 片方はアリス、あなたそっくりだけどもう一方は、後ろ姿からするとわたしに似てなくもないわね」とか？

だがあの場面ではそんな質問を思いつきもしなかったし、アリスが出し抜けに

「魔理沙が来るの――いまから、ここに」

と宣言したことでそんな疑問も消し飛んでしまったのだった。

3.2 解析学からの無限小概念の追放

魔理沙は今回も空堀づたいに図書館に入ってきた。もうお昼も近いというのに「おはよう」などと、わたしたちのどちらにともなく挨拶してからこちらに向き直ると早速要件を切り出した――。

「ねえパチュリー、IST っての教えてよ」

「IST って言っても色々あるわね。ぱっと思いつくだけでも Intuitionistic Set Theory (直観主義的集合論) もあるし、Internal Set Theory (内的集合論) もあるし、Irish Standard Time (アイルランド標準時) なんかもあるわ」

「あ、それぞれ」

「《それ》じゃわからないでしょ」

「Internal Set Theory のほう」

「ねえ魔理沙、この IST が何をやるものなのかは知ってるの？」

「知ってるよ、超準解析でしょ？」

「ええその通りよ。では超準解析は何をやるものかしら？」

「超準解析は微積分の議論をすごく簡単にしてくれるでしょ」

乱暴な要約だが、まったく間違っているとも言えない。それにしてもどうして魔理沙が超準解析に興味を持っているのだろうか。

「なるほどね、では魔理沙の言うとおりの微積分の議論が超準解析によって簡単になるとしたら、そもそもなんで微積分の議論は面倒になってしまったのだと思う？」

「んん――なんでだろう？」

ふとアリスのほうを見ると、彼女はただ静かにお茶を飲んでいた。魔理沙とわたしの会話に参加するつもりはなさそうだ。彼女の視線の先にあるミニチュアテーブルには、さらに一体の人形――魔理沙の姿の人形――が出現している。魔理沙の人形とわたしの人形はなにか話し合っている素振りをみせて小刻みに動いていた。