

# 目次

第 0 章	まえがき	@xhl_kogitsune	ii
第 1 章	クラスタリングしなイカ?	@dekosuke	1
1.1	アルゴリズムとライブラリを決めなイカ? . . . . .		1
1.2	CSV をパースしなイカ? . . . . .		3
1.3	実際にやってみるでゲソ . . . . .		3
1.4	まとめでゲソ . . . . .		5
1.5	参考文献でゲソ . . . . .		5
1.6	kmeans-vector の license でゲソ . . . . .		6
第 2 章	Copilot への希望と絶望の相転移	@master_q	7
2.1	Copilot はイカが? . . . . .		7
2.2	Copilot DSL と C 言語コードの融合でゲソ . . . . .		11
2.3	Fibonacci.hs を例にして Copilot DSL の書き方を伝授するでゲソ! . . . . .		11
2.4	Wii リモコンで遊ぶでゲソ . . . . .		13
2.5	libc のないところでも操縦できるでゲッソ! . . . . .		16
2.6	けっきょく使えそうなんでゲソ? . . . . .		21
第 3 章	加速しなイカ?!	@nushio	23
3.1	これまでのあらすじ . . . . .		23
3.2	進化しなイカ? . . . . .		23
3.3	生存戦略しなイカ? . . . . .		27
3.4	種をまかなイカ? . . . . .		29
3.5	統計しなイカ? . . . . .		31
	参考文献 . . . . .		32
第 4 章	Java VM 上で tail call しなイカ?	@yryozo, @xhl_kogitsune	33
4.1	Java VM と関数型言語は相性が悪いでゲソ? . . . . .		33
4.2	Java VM 上の関数型言語実装を紹介するでゲソ! . . . . .		38
4.3	Java VM 自体も侵略するでゲソ . . . . .		43
	参考文献 . . . . .		52
第 5 章	関数型イカガール第 3 話	@tanakh	55
第 6 章	映画けいさん!	@tanakh	61
	会員名簿じゃなイカ?		76

## 第0章

# まえがき

— @xhl\_kogitsune

### 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. まさかの二冊目の本ですが。

B. わけがわからないよ。

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! イカ娘」に続く、二冊目の関数型イカ娘の本でゲソ。アニメ 2 期も執筆時点で放送中だし、いいタイミングじゃなイカ!?

関数型言語で地上を侵略しなイカ!

## 第1章

# クラスタリングしなイカ？

— @dekosuke

最近、ついったーのタイムライン・・・ゲフンゲフン、美しい海を見ていると、〇〇クラスタという単語をよく聞くでゲソ。例えば「関数型クラスタ」というと、関数型言語が好きな人たちの集まりを指すでゲソ\*1。一般に、クラスタリングとは集団を似た者同士の小さなグループに分けることを言うでゲソ。クラスタリングは、その集団の性質を知ることにつながるでゲソ。言い換えれば、クラスタリングしてしまえばその領地を侵略したも同然でゲソ！！

というわけで、クラスタリングをやってみなイカ？

### 1.1 アルゴリズムとライブラリを決めなイカ？

やることは決まったので、クラスタリングのライブラリを探すでゲソ。Hackage のパッケージリスト <http://hackage.haskell.org/packages/archive/pkg-list.html>

を、“Clustering”で検索すればいいんじゃないイカ？6件ぐらいヒットするんじゃないイカ？

このうち、“hierarchical-clustering” (階層クラスタリング) 系のライブラリは興味深いけど話が複雑になるので今回は避けるでゲソ。hsgsom (自己組織化マップ) は高精度のクラスタリング手法でゲソが、ライブラリ作者がやる気を出しすぎて扱いが大変なので避けるでゲソ。

そして、残った希望、いやライブラリは・・・K-Means のライブラリでゲソ。K-Means はクラスタリングにおける一番シンプルなアルゴリズムでゲソ。ソートで言うとかイックソートみたいなアルゴリズムでゲソ\*2。

というわけで、kmeans ライブラリをインストール・・・するのが許されたのは 2011 年 10 月頭まででゲソ。なぜベストを尽くさなイカ？2011 年 10 月に kmeans より効率的な kmeans-vector ライブラリが作られたでゲソ\*3。List を使っている kmeans ライブラリと違い、kmeans-vector ライブラリは Data.Vector\*4を使っているので4倍高速\*5でゲソ！！！！

```
cabal install kmeans-vector
```

で、必要なものはインストールできたんじゃないイカ？え、ライブラリの内容が分からないでゲソか？大丈夫、K-Means は簡単なライブラリなので、中身を全部読めるでゲソ。短いのでここに転載してもいいんじゃないイカ？\*6

\*1 もちろん私も関数型言語クラスタに含まれているでゲソ

\*2 K-Means に関しては計算量がデータに対して NP 困難なので、大規模なデータに対しては近似手法が取られるでゲソ。今回はそんなことは気にしないことにするでゲソ

\*3 ちなみに Hackage のパッケージリストには載ってないでゲソが、理由は筆者の知識では分からないでゲソ

\*4 vector はランダムアクセスが定数時間で行えるデータ構造でゲソ。数値計算の人が使うのはだいたいランダムアクセスでゲソ

\*5 kmeans-vector ライブラリ作者調べ

\*6 kmeans-vector のライセンスは BSD3 に準じるでゲソ。章末に記載するでゲソ。

## 第2章

# Copilotへの希望と絶望の相転移

— @master\_q

みんないつも参照透明な海を泳いでるでゲソ？海のみぐみは大きいでゲソ。素晴らしいでゲソ。でもそのみぐみは大きな力によって保たれているでゲソ。例えば簡単な Haskell 製のプログラムを `ghc` でコンパイルしてみるでゲソ。

```
$ ldd HaskellApp
linux-vdso.so.1 => (0x00007fff4ad7c000)
libgmp.so.10 => /usr/lib/x86_64-linux-gnu/libgmp.so.10 (0x00007f250dd37000)
libffi.so.5 => /usr/lib/x86_64-linux-gnu/libffi.so.5 (0x00007f250db2a000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007f250d8a7000)
librt.so.1 => /lib/x86_64-linux-gnu/librt.so.1 (0x00007f250d69f000)
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f250d49b000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f250d116000)
libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007f250cefa000)
/lib64/ld-linux-x86-64.so.2 (0x00007f250dfd6000)
```

依存ライブラリが多いでゲソ、、、特に厳しいのが、`libgmp`、`libc`、`libpthread` あたりでゲソ。これらのライブラリに強く依存していてなかなかこれらのライブラリのない環境では参照透明な海のみぐみを受けられないでゲソ。。。

### 2.1 Copilot はイカが？

そんなことに悩んでいたおり

Copilot <<http://leepike.github.com/Copilot/>>

を使うと Haskell の DSL で書いたプログラムが C 言語に落ちるという情報を `tanakh` 殿から仕入れたでゲソ。Copilot を使えば、Haskell 製プログラムから `libgmp`、`libc`、`libpthread` などへの依存をなくせて、左記ライブラリのない組み込み用途の CPU でも Haskell プログラミングできるんじゃないイカ？ちょっと使ってみるでゲッソ！使い方は [Copilot tutorial](#) \*1 が詳しいでゲソ。

Copilot のしくみを簡単に解説すると、Copilot はある種の DSL で、

1. Copilot 外部の変数や関数を呼び出して、`Stream` という型に収め
2. `Stream` をフィルタしたり `Stream` 同士を演算したりして
3. 条件によって外部へ副作用を与える (`trigger`)

\*1 [https://github.com/niswegmann/copilot-discussion/blob/master/tutorial/copilot\\_tutorial.pdf](https://github.com/niswegmann/copilot-discussion/blob/master/tutorial/copilot_tutorial.pdf)

## 第3章

# 加速しなイカ？！

— @nushio

### 3.1 これまでのあらすじ

■**夢のようなライブラリがあったじゃなイカ！** これからは並列計算の時代だと聞いて闇の言語侵略に闘志を燃やす関数型イカ娘。そこにキュウベエアイコンの人が現れて、避けようのない綻びも驚きもすべて君が抽象化してしまえば良い、そのための力が Haskell には備わっているし、彼女も出来ると説くのだった。

■**GPU も、FPGA も、あるでゲソ！** イカ娘は Accelerate をつかった GPGPU コード生成を試してみる。Accelerate ちゃんは、これからの並列計算のコードを生成しまくっちゃいますからねーと意気込むのであった。

■**実アプリの性能と向きあわなイカ？** 緑色っぽい CUDA 使いに性能の差を見せつけられた Accelerate ちゃんは、存在意義を見失い、その魂を黒く黒く染め上げていくのだった。

■**そんなの、私が許さないでゲソ！** イカ娘は涙ながらに、自分自身の手でちゃんと速度の出るコードを生成してみせることを誓うのだった。いいよ、Accelerate ちゃんのためなら、私・・・。

■**最後に残った半蔵門線でゲソ！** ついに東京をワルブルギスの夜が襲う。台風により交通網が麻痺する中、イカ娘たちは ICFP 会場から PFI 社までサイモン先生を護送する必要に迫られていた。

— そして、現状はというと・・・

### 3.2 進化しなイカ？



「フーハハハハハハ！！数々の未来ガジェットを生み出してきた我がラボの最新作が完璧に完成したでゲソ！」

— イカちゃんは時間遡行者の真似をしている。個人的には、そっちじゃないほうの時間遡行者が好きなのだが。



「ぼやいたって仕方がないでゲソ。さあ、本題に行かなイカ？この章では、拙作の偏微分方程式の陽的解法コードを GPU その他並列計算機向けに生成するためのドメイン特化言語、Paraiso(<http://paraiso-lang.org/wiki/>)を扱うでゲソ。Paraiso の実装の詳細については、函数プログラミングの集い 2011 などでも紹介したから説明を省略するとして、今回はそこからのアップデートをご紹介するでゲソ。なんと、Paraiso には遺伝的アルゴリズムベースの自動チューニング機構が備わったのでゲソ。すごいじゃなイカ！これで、すべての偏微分方程式の陽解法を速くできるでゲソ。過去と未来、すべてのアーキテクチャで、すべてのコードを、この手で！」

— と言いたいところだが、今のところ対応しているのは OpenMP もしくは CUDA のみなのは相変わらずだ。それにしても、Paraiso の詳細を完全に飛ばしたら、さすがに以後の説明に差し障るのじゃなイカ。三行くらいでまとめられなイカ？



「だるいでゲソ！それに三行じゃとても足りないでゲソ！」



「私からもお願いできるかしら」

## 第4章

# Java VM 上で tail call しないイカ？

— @ryozo, @xhl\_kogitsune

お前たち、海の中でも海の家でも陸の上でも、Linux の上でも FreeBSD の上でも AIX の上でも x86 の上でも Power の上でも、できることならいつでもどこでも関数型言語を使いたいと思うはずでゲソ。でも、言語処理系をマルチプラットフォーム対応にするのは大変な労力でゲソ。そこで、Java VM 上で関数型言語を実行したり実装したりしないイカ!?

### 4.1 Java VM と関数型言語は相性が悪いでゲソ？

最近 Java VM の上でも関数型言語を走らせている人が多くなってきているでゲソ。特に Scala や Clojure なんかが人気が高いようでゲソね。それもこれも私の侵略計画が順調に進んでいる証拠じゃないイカ？

Java VM の上での実装には、おおまかに 2 つの方法があるでゲソ。

**インタプリタ** Java VM 上で実装したい言語のインタプリタを実行する方式でゲソ。

**トランスレータ** 実装したい言語のソースコードを、Java VM 上のプログラム (Java バイトコード) に変換し、その変換した Java バイトコードを Java VM 上で直接実行する方式でゲソ。実装したい言語の関数が、Java のメソッドに変換されるものが多いでゲソ。

ところが、Java VM と関数型言語 (のトランスレータ方式の実装) は相性があまりよくないという人たちもいるでゲソ。というのも、関数型言語に必須と言ってもいい末尾呼び出しの最適化 (Tail Call Optimization, TCO) が Java VM ではすごくやりにくいのでゲソ。理由は下の表に書いた通りでゲソが、Java VM にはスタックフレームを積んだりポップしたりせずにメソッド外に飛ぶという命令がないのでゲソ<sup>\*1</sup>。TCO がないと、Scheme で末尾再帰や tail call でループ書いただけで即死でゲソ!

分類	命令名
メソッド外に飛べるが、スタックフレームを積むかポップしてしまう	invokevirtual, invokestatic, invokeinterface, invokespecial, return, areturn, ireturn, lreturn, freturn, dreturn, athrow
スタックフレームの深さは変えないが、メソッド外には飛べない	if_acmp, if_icmp, if, ifnonnull, ifnull, tableswitch, lookupswitch, goto, goto_w, jsr, jsr_w, ret

<sup>\*1</sup> ちなみに、この問題は Java VM だけのものではないでゲソ。例えば関数型の言語を C にトランスレートして gcc なんかでコンパイルしようとする時にも同じ問題が起こるでゲソ。

## 第5章

# 関数型イカガール第3話

— @tanakh

この作品はフィクションであり、実在する人物、地名、団体とは一切関係ありません。

---

12月。冬の海に僕はいた。夏の賑わいは見る影もなく、寂しい海にはもちろん他には誰もいない。海の家は夏の終わりと共に閉店し、あのイカの少女も姿を消した。

それでも僕は週末が来るたびここに足を運ぶ。あの時の胸の高鳴りを忘れることができなくてー

---

最近の僕を悩ませているのは、圏論だ。正確には圏論そのものではなく、代数学とプログラミングの関わりについてだ。圏論のことを考えているときはとても楽しくて、それだけで時間を潰してしまうことも少なくないのだけでも、プログラミングの時にどことなく圏論の影がちらつく Haskell は勉強を始めた当初は手こずったりしたものだった。それが Haskell、あるいは関数型パラダイムの敷居を上げているのではないか、少なくとも心理的障壁になっているのではないかと感じてしまう。

「そんなの決まっているじゃなイカ！」

突然の声に僕は顔を上げた。そこにはあのイカの少女のくりくりとした大きな目があった。彼女は夕日を背に、僕のノートを覗き込んでいた。いつからそこにいたのだろう。どうやら僕は考え事をするとうろたえなくなるようだ。しかしそんなことよりも、イカ娘にまた会えたことが嬉しかった。

「それは Composability のためでゲソ」

そんな僕にはお構いなしにイカ娘は続ける。

「Composability はプログラムを自由に組み合わせられる性質でゲソ。プログラムの Composability は、安心してプログラムを書くためにとても重要なことでゲソ。基本的なことじゃなイカ」

Composability という概念は聞いたことがないでもなかったが、それがそんなに重要なものだとはいえなかったわけ。

「コンビネータ、って聞いたことないでゲソか？人間の世界では小学校で習うって聞いたでゲソが」

はて、どこの世界の小学校の話だろうか？

「たけるがドラゴンの表紙の分厚い本を教科書に使ってるでゲソ。確か、ドラゴンなんか<sup>\*1</sup>とか…」

どういう小学校<sup>\*2</sup>なんだ。それはともかく、コンビネータっていうのは自由変数を含まないラムダ式のことだ。狭義には関数を引数に取って、それを組み合わせて新しい関数を返すような高階関数を指す

---

<sup>\*1</sup> 「コンパイラー原理・技法・ツール」通称“ドラゴンブック”

<sup>\*2</sup> Functional Ikamusume Advent Calendar 2011 一日目「たけるの通う小学校に Dragon Book や SystemC 等の専門書籍がてんこもりでヤバイでゲソ！」<http://d.hatena.ne.jp/xiaohuli3/20111201#1322758513> 参照

## 第6章

# 映画けいさん！

— @tanakh

この作品は某映画を見たいけどなかなか見に行けない私の妄想の産物、つまりフィクションであり、実在する人物、地名、団体とは一切関係ありません。

---

「唯先輩、起きてください」  
 「むにやむにや、あと1時間だけ……」  
 「ダメです。そんな時間ありませんよ」  
 「むにやむにや、ハスにやん。もう食べられないよ」  
 「何寝ぼけてるんですか。着きましたよ、ロンドン」  
 「……あ、おはよう、ハスにやん」

---

私たちは今、ラムダ高計算機プログラミング競技研究部、通称「計算部」の課外活動としてイギリスに来ています。

飛行機の中でずっと私にもたれかかって寝ていたのが、平無駄唯先輩。いつもはとぼけているように見えて、今私たちがここに来ているのも唯先輩のおかげだったりします。

「ロンドンだー、ロンドンだー」  
 「ヒースローだー！」

あそこで唯先輩と一緒にしゃいでいるのが、排中律先輩。2年前に潰れかけだった計算部を復活させた人で、計算部の部長です。普段はいい加減な人ですが、いざというときは…やっぱり頼りないかも。

「おい、濡。何してんだ。早く荷物取ってこいよ」  
 「うう…飛行機怖い……」  
 「飛行機はもう降りただろ。おい、しっかりしろー」  
 律先輩の隣で青い顔をしているのが、入文字山濡先輩。計算部で唯一の常識人で、実力もトップクラス。私のあこがれの先輩です。怖がりなのが玉に瑕かな。  
 「あらあら、ハスさちゃんは常識人じゃないのかしら？」  
 「あつ、モナ先輩。なんで私の考えてることを！？」  
 そしてこれが、型吹モナ先輩。おっとりしていて、すごいお嬢様なんだけど、ちょっと何を考えているのかわからない所があったりします。  
 「はあ、こんなので大丈夫なのかなあ」  
 海外に来て、計算部は相変わらずです。  
 「ハスにやん、すごいよ！白人がいっぱいいるよ！」