



Me and Ope:
HIMA#4

Jan 25

第4回 HIMA(Haskell Internet Meeting Anytime) 開催地

HIMAはHaskellプログラミングに興味をもつ人が、とりとめなく、おしゃべりをするという会です。一応のお題はありますが、Haskellプログラミングの知識経験の有無深浅にかかわらず、興味のあるかたはご参加ください。もちろん、ROMあり、茶々あります。今回は以下の要領で開催します。

- 日時： 2010年1月23日(土曜日) 20:00~23:00
- 場所： ここ
- 幹事： 酒井
- お題： Functional Reactive Programming (FRP)

参考:

- <http://d.hatena.ne.jp/maoe/20100109/1263059731>
- <http://d.hatena.ne.jp/maoe/20100116/1263661213>
- <http://d.hatena.ne.jp/maoe/20100123/1264225964>



Maoe and Ope: こんにちは。テストテスト。

Jan 25



Nobsun and Ope: waveって It's All Text!で外部エディタを使って書き込みとかできないんだっけか。

Jan 25



豊福 and Ope: まずガウスのように始めたまえ。

Jan 25



豊福 and Ope: 書き込み方わからん。これ書けてるのかな？

Jan 25



豊福 and Ope: かけてますかどうも。

Jan 25



豊福 and Ope: 今から呑みにでかけます。ログだけ取るままにして出かけます。では。

Jan 25



Tadashi Hirata and Ope: うーん、できないっぽいなあ >外部エディタ

Jan 25



Me and Ope: 時間ですね。はじめましょうか。

Jan 25



Kazu and Ope: こんにちは。

Jan 25



Me and Ope: こんにちは。Google Wave はまだそんなに使えてないので、どんな感じになるか分からないのですが、よろしくお願ひします。

Jan 25

今回はFRPがテーマということで、maoeさんの記事をネタに色々語れたらと思います。

  **Me and Ope:** maoeさんいるかな? Jan 25
私はmaoeさんの記事は、あらかじめ読んでおこうと思いつつも読めてなくて、今読んでます。

  **Maeo and Ope:** 今来ました Jan 25

  **Me and Ope:** きたー Jan 25

  **Me and Ope:** まず、Reactive Programming とは何かというところを読んです Jan 25
が、Excelが実はreactive programmingだというのは言われて目から鱗でした。

  **Maeo and Ope:** Excelをつかったreactive programming Jan 25
は<http://d.hatena.ne.jp/NyaRuRu/20080317/p1> こちら辺に書いてあるのを見てほほーと思って書きました

  **Me and Ope:** おお。これはすごいですね。 Jan 25

  **Maeo and Ope:** reactive programmingとか言われても何に使うんだろうと思ってたんで Jan 25
すが、ああなるほどなと思いました。

  **Maeo and Ope:** ただ、これを見てもHaskellでどんな感じになるのかはさっぱりわかり Jan 25
ませんでした。

  **Me and Ope:** たしかに。 Jan 25
で、それをHaskellに持ってきたのが FRP だと。

  **Maeo and Ope:** そうですね Jan 25
それで、たぶんこれを使うと、IOモナドだらけのプログラムをもう少し関数型の人でも気持ちいい形で書けるようになるんじゃないかなと期待しています。
reactiveというからには外部から何か入力を受けて、それに反応するプログラムが向いているわけですが、よくよく考えるといわゆるサーバといわれるものはたいていそういうプログラムなので、サーバを綺麗に書くのに有効なのかなという印象です。今のところ。

  **Kazu and Ope:** 質問 Jan 25
maoe さんの二番目の記事の
`bellEvent = beepAction <$> beepTimer`
は
`bellEvent = bellAction <$> beepTimer`
の間違いですか？
あれ、bell とbeep が全体的に揺れているなあ。
この辺り、全部見直す必要があるかも。

  **Maeo and Ope:** 間違いですね。直しておきます。ありがとうございます Jan 25
実際に動くコードは<http://github.com/maoe/reactive-examples> をどうぞ
名前をあとから変えた影響が。。。

  **Maeo and Ope:** 表記揺れはたぶん無くなりました Jan 25

  **Me and Ope:** おお、Applicative 便利だ。 Jan 25
一度 Event (IO a) にして、そこから IO a にするのが、なるほどという感じ。
Behaviorは連続的に変化するからこういうことは出来なさそうだけど、Eventは離散的に発生するイベントだから、こういうことが出来るんですね。

reactiveは以前に論文を読んだっきりだったんですが、やっぱりよく考えられてるなあ。



Me and Ope: Yampa派(?)としてはどうでしょう? > shelarcyさん

Jan 25



Shelarcy and Ope: 私が Yampa 派だったのは Conal の reactive パッケージが出

てくるまでだったので、特に Yampa 派というわけではないです。Yampa は性能面の問題が指摘されているので、その辺で Conal の reactive や Grapefruit の方が強い感じですね。(Yampa に対する性能向上を図ろうという試みもあったのですが、その後続く論文や実装が存在しないので.....。)性能とプログラミングにおける使いやすさ二つが両立できるのであれば、Yampa をやめて rective 等へ乗り換えても良いかなと思います。(現在問題があって Conal は wxHaskell を使っていませんが、問題を解決し次第 wxHaskell 上に FRP な GUI ライブラリを作ると Conal も言っていますし.....。)



Me and Ope: Yampaに対する性能向上だと、一応 Causal commutative arrows and their optimization みたいなのはあるかと。

<http://portal.acm.org/citation.cfm?id=1596550.1596559>

<http://www.cs.yale.edu/homes/hl293/download/icfp066-liu.pdf>



Shelarcy and Ope: その成果が 0.9.2 以降の Yampa パッケージに取り込まれて

いるのですが、まだまだ性能面での問題が残っていて、そこがまだ改善されていないという意味です。現在の Yampa



Shelarcy and Ope: あー、酒井さんの示してくれた論文の方の最適化は

まだ取り込まれていませんね。別の論文と勘違いしていました。ただ、いずれにせよ、HackageDB の方にあるパッケージに成果が取り込まれないとユーザーが恩恵を得られないので.....その点で、Conal が積極的に関わっている reactive パッケージの方が性能的に有利だと思います。

パッケージ (のソースコード) をダウンロードすれば、その論文の成果が取り込まれていることとまだまだ改善の余地があることが (コメントなどから) 分かると思います。

<http://hackage.haskell.org/package/Yampa>



Maoe and Ope: memory leakとかtime leakを解決するためにarrowisedなFRPが

できて、その成果物としてyampaがあると思っていたので、yampaは速いのだと思っていました。性能面に問題があるとは。。。



Shelarcy and Ope: 処理が動的になってしまって、静的な最適化が効かな

いので「もっと静的にして、静的な最適化ができるようにする!」という趣旨の話がこちらの論文に書いてありますね。 <http://www.cs.nott.ac.uk/~nhn/Publications/icfp2005.pdf> スライド: <http://www.cs.nott.ac.uk/~nhn/Talks/ICFP2005-DynamicOptimizationUsingGADTs-4up.pdf>



Me and Ope: そうだったのか。僕も結構意外です。

Jan 25



Maoe and Ope: Behaviorはsnapshotで某かのEventでサンプリングしてEventに変えてやる必要があるみたいです。



Me and Ope: ふむふむ

Jan 25



Me and Ope: あと、FRPにはpushとpullがあると。

Jan 25

pullというのは一定時間ごとに出力値を再計算しようとして、トップダウンに動かしていく感じが

な。
で、pushは逆に入力側から考えて、入力に変化したらそれに影響をうける部分を再計算するという感じかな。

  **Maoue and Ope:** pullの方が関数的なスタイルになるのが良くて、pushの方が性能がよい（レイテンシが少ない）のが利点と書いてありましたが、push式は触ったことがないのでどんなモンなのか気になるところ

  **Me and Ope:** どっちかという実装の詳細のような気もするけど、プログラミングスタイルへの影響もあるのでしょうか？

  **Maoue and Ope:** reactiveの論文にはスタイルに違いがあるようなことが書いてあったと思います

  **Kazu and Ope:** FRP で web サーバーを書いたとします。GET イベントを受け、ファイルを読んで返す関数の型は、どうなるのですか？ IO a から逃れられますか？

  **Maoue and Ope:** Event Request -> Event Responseな関数とEvent Response -> Event Actionな関数を作って組み合わせる感じになるのではないかなと思います。前者ではIOができないので完全なResponseが作れなような気がしますが。。。HTTPサーバとかもっと簡単どころではircボットとかを作ってみるのがよいのかなあ。

  **Ltakeshi and Ope:**

  **Nobsun and Ope:** ただいま自宅に帰りました。亀コメントですが、Haskell はIOからはのがれられないですねえ。Haskellであるかぎり。

  **Maoue and Ope:** IOから逃れることはできなそうだと思います。できなそうなのですが、Event (IO ())なイベントとしてIOを扱うことで、IOべったりなコードよりは納得できるスタイルで書けるんじゃないかなあと思います。実際に大きな例をかいてないのであれですが。

  **Me and Ope:** 単発の処理だとあまり嬉しさはないのかなという気が。それよりは入力値の変化やイベントによって状態が変わったりとか、そういうやつの方が嬉しそう。

  **Shelarcy and Ope:** ところで、現在開催中の POPL の併設 Workshop（または Symposium）の FRP 関係の論文を読んでいる方はいますか？

  **Shelarcy and Ope:** 例えば、"Mixed-level Embedding and JIT Compilation for an Iteratively Staged DSL" <http://www.cs.nott.ac.uk/~ggg/publications/wflp2010.pdf>

  **Shelarcy and Ope:** Paul Hudak と共著の論文も FRP 関係だと思うのですが、そちらの方は Web からでは見られないようです.....。

  **Me and Ope:** 論文タイトルは何ですか？

  **Shelarcy and Ope:** 共著者の Hai Liu のページに論文が出ていました。最新の Jan 25 論文二つです。 <http://www.cs.yale.edu/homes/hl293/>が、FRP 関係ではなかったっ ぽいですね。

  **Maoe and Ope:** FRP関連の論文といえば、Safe Functional Reactive Jan 25 Programming through Dependent Typesというのを見かけて興味がわいています。reactive でプログラムを書いているとメモリリークはわりとすぐ起こるので、そういうのが無くなるのかなあ と。

  **Me and Ope:** ICFP2009の論文ですね。それは私も読もうと思ってました。 Jan 25

  **Me and Ope:** え、それFRP関係の論文なの? Jan 25
ぜんぜん追いかけてないのだけど、タイトルからはあまりそうは見えないような。

  **Shelarcy and Ope:** みたいですよ。著者の一人が、YampaSynth という Yampa Jan 25 でシンセナイザーを作成した人で、もう一人が Henrik Nilsson だったので FRP 関係の論 文だと思って見てみたら、Fuctional Reactive Programming という単語が論文中にありました。 (きちんとは読んでいませんが。) <http://hackage.haskell.org/package/YampaSynth>

  **Me and Ope:** The FHM design was originally inspired by Functional Reactive Jan 25 Program-
ming (FRP) [4], particularly Yampa [12]. A key difference is that FRP provides *functions* on signals whereas FHM generalises this to *relations* on signals.
とか書いてありますね。制約プログラミングっぽい感じの書き方が出来るのかな。

  **Me and Ope:** 話を戻すと、.....ええと、何だっけ? Jan 25
とりあえず、Behaviorの説明を読んです。

  **Kazuhiko Sakaguchi and Ope:** 来てみました。テスト Jan 25

  **Me and Ope:** いらっしゃーい。 Jan 25

  **Maoe and Ope:** 実際にFRPのプログラムを書くときにBehaviorがうまくはまる事例がし Jan 25
りたいなあ

  **Shelarcy and Ope:** Event と Behavior という区切りが正しい物であるかどうか Jan 25
は分かりませんが、Yampa の何でも Event というスタイルでは、実際に使ってみて、どの 関数がどういう意図で使用されているかがかなり不明確で、すでに書かれたコードの意図が読み取り難 いという印象を受けました。なので、そういう意図の区別がつけ難い所は、何らかの形で区別できるべ きだと思いますね。

  **Me and Ope:** ん? YampaのSFって意味的には Signal → Signal で、Signal = Jan 25
Behavior だから、なんでもEventというのは違うのでは?

  **Shelarcy and Ope:** うーん.....実際に書かれたコードの意味が取り難かったの Jan 25
で、そこがどうにかならないのかという意味です。もちろん、きちんと考えればそれぞれの 関数の意味を確かめることができるのですが、Yampa のスタイルではソースコード上の意図を読み解く 難易度が高かったんで.....もちろん、慣れの問題もあるでしょうけれど。



Me and Ope: いや、そもそも「何でもイベント」なんですか？

Jan 25



Maeo and Ope: Conalがarrowベースの方はeventとbehaviorの区別が無くなるのがいやだとしていたのを見かけました。全部eventなのかわかりません。



Me and Ope: ああ、分かった気がする。
SF a b が基本で、イベントを扱うときは a, b のところを Maybe にするんですけどっけ？

Jan 25



Shelarcy and Ope: そんな感じです。 Maybe 型の代わりに data Event a = Event a | NoEvent という型を使いますが。

Jan 25



Me and Ope: Yampaでの事例をReactiveで書くことを考えると、Behaviorは自然に出てくるんじゃないかな。

Jan 25

- Arrows, Robots, and Functional Reactive Programming
<http://www.haskell.org/yale/papers/oxford02/index.html>
- Functional Reactive Robotics: An Exercise in Principled Integration of Domain-Specific Languages
<http://haskell.cs.yale.edu/yale/papers/ppdp02/index.html>

あたりとか。



Maeo and Ope: integralを見て思い出したのですが、Behavior編のIntegrall.hsというサンプルコードはtime leakするのですよ。それがintegralのところの原因ぽいのですが、どう直したらいいのかさっぱりわからないのです。

Jan 25



Me and Ope: 質問: time leak って何ですか？

Jan 25



Maeo and Ope: 実行するとよくわかるのですが、徐々にCPU使用率が上がって重くなっていくんです。

Jan 25



Me and Ope: 試してみよっと。
まずは cabal install reactive

Jan 25

あれ、少し試してみたけど、CPU使用率は特に上がってこないです。
Windows 上の GHC 6.10.1 で `ghc --make -O Integrall.hs` でコンパイルして、実行して、キーを適当に連打しまくってみました。



Me and Ope: って、あれ、そもそも入力したキーがprintされなくちゃいけないのかな。何も出力されてないや。

Jan 25



Maeo and Ope: 入力はhSetEchoで消しているのが正常な動作です。わかりにくいから表示するようにした方が良さそうですね。

Jan 25



Me and Ope: でも、`adaptE $ print <$> ...` ってあるから、何かprintされるのかなと。

Jan 25



Maeo and Ope: あれ？ホントは毎秒数字が出てくるはずですよ。このサンプルプログラムは>とか<を押すと速度を変えることができ、その速度で走っているときの位置というか距離を毎秒出力するようになっています。

Jan 25



Me and Ope: 何も出てきてないです。

Jan 25

 **IKEGAMI Daisuke and Ope:** Integral.hs はどこにありますか？ Jan 25
cabal install reactive はしました。すばらし、ありがとう。こんばんは、みなさん。こちらの環境では、Integral.hs は 2 秒おきに、数字(Double)が増えて表示されています。leak しているのかどうかは、これから観察します。コンパイル時に最適化するかしらないかで違いますね。-O つけなかったら 100% CPU 食うけど、付けたら多少はましになりました。

 **Me and Ope:** こんばんは— Jan 25
池上さんはMacでしたっけ？ じゃあ、Windows固有の問題かなあ。
Linuxで試すか..... Linuxだと確かにそうなりますね。

 **IKEGAMI Daisuke and Ope:** はい、Mac です。 Jan 25

 **Maoe and Ope:** こんばんは。http://github.com/maoe/reactive- Jan 25
examples のIntegrall.hsです。

 **Maoe and Ope:** 2秒おきというのもおかしいな。僕の手元では-O Jan 25
の有無にかかわらず重くなっていきます。

 **IKEGAMI Daisuke and Ope:** [0.2..] なんだから、いいんじゃない Jan 25
でしょうか。違うのかな。

 **Maoe and Ope:** Oh! Integral.hsではなくてIntegrall.hsの方です。 Jan 25
Integralの方もリークしているとは！

 **IKEGAMI Daisuke and Ope:** Integral.hs のほうは -O したら、 Jan 25
leak してなさそうです。Integrall.hs のほうもこれから見てみます。
Integrall.hs も -O つけたら特に time leak というものはなさそうです。上がったりがったり操作できて楽しいね。
勘違い、'0' は上下を止める操作なんだ。

 **Maoe and Ope:** お、リークしてないですか。やはり環境依存な Jan 25
のかなあ。
そうです > 上下を止める操作

 **Maoe and Ope:** あれれ。環境依存でしたか。僕はLinux上のGHC 6.12.1 Jan 25
です。

 **Shelarcy and Ope:** ところで、どうでも良い点かもしれませんが、みな Jan 25
さんがつけている最適化オプションは -O ですか？ -O2 ではなくて。

 **Me and Ope:** 僕は -O ですね。 Jan 25

 **Maoe and Ope:** ロボット制御はなかなかぴんと来ないけどやってみよう。 Jan 25

 **Me and Ope:** あと、去年と一昨年 of ICFPC の問題を、FRP でちゃんと解いてみ Jan 25
ようと思ってたのを思いだした。
特に一昨年 of 問題 <http://web.cecs.pdx.edu/~sheard/2008lcfpContest/> はセンサーデータとかイベント
の処理や制御と、プランニングの処理とを書く必要があるので、そのあたりをFRPを使ってうまく整理して書けるのではないかと期待してます。

 **Maoe and Ope:** たしかFRPのフレームワークを自作されたとおっしゃってましたよね？ Jan 25

 **Me and Ope:** 去年のやつは、Arrowベースの簡単なものを自作してやりました。 Jan 25

 **Maoe and Ope:** ArrowベースというとやっぱりYampaみたいにSFなんですか？ Jan 25

 **Me and Ope:** そうです。 Jan 25
<http://www.tom.sfc.keio.ac.jp/~sakai/d/?date=20090630#p01>

SFと名前は違うけど、こんな型でやってみました。

-- Mealy Automata

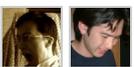
newtype Automata i o = Automata { next :: i -> (Automata i o, o) }

 **Maoe and Ope:** Arrowベースとreactiveみたいな古典的なやつとで、書きやすさの違いとかってありますか？ Jan 25

 **Nobsun and Ope:** 単なる予想です(書いたことがないので)がArrowは2引数なので、その分難しいかと。 Jan 25

 **IKEGAMI Daisuke and Ope:** Arrow notation 面白いし、いいなーとは思いますが、うんですけどどうでしょうか。 Jan 25

 **Maoe and Ope:** Conal Elliottはやだーってってますね Jan 25

 **IKEGAMI Daisuke and Ope:** 誰の論文だったか、はっきり覚えていないのですが、「Arrow は theoretical に遅くなることがわかるので避ける」とか、えーと、Parsec だったかな。なぜ monadic parser combination にするのか、というところで触れた。で、その理由はよくわかりません。 Jan 25

 **Me and Ope:** Arrowの最初の動機のひとつにMonadベースのパースアの効率の悪さがあったのに、それが本当だとしたら皮肉だなあ。 Jan 25

 **Maoe and Ope:** ふむ。そんなことがあるのですか。よく見たら <http://conal.net/blog/posts/why-classic-FRP-does-not-fit-interactive-behavior/> ここにもArrowだと非効率みたいなことが書いてありました Jan 25

 **Nobsun and Ope:** なぜ非効率なのでしょう。 Jan 25

 **Me and Ope:** 私も気になります。 Jan 25

 **Maoe and Ope:** Jan 25

 **Me and Ope:** 「Arrowの最初の動機のひとつにMonadベースのパースアの効率の悪さが」というのは Generalising Monads to Arrows に載ってます。 Jan 25

  **IKEGAMI Daisuke and Ope:** 著者の John Hughes を問いつめよう、残念ながら日本に来る機会はそうそうないんだけど Jan 25

  **Nobsun and Ope:** 4月にはこないのかしらん? Jan 25

  **IKEGAMI Daisuke and Ope:** 4月になにがあるんだっけ、AIMXIに誰が来るのかはまだ聞いてないです。FLOPS 2010 (Apr 19-21) には来ないと思う Jan 25

  **Nobsun and Ope:** それはざんねん Jan 25

   **Me, IKEGAMI Daisuke and Ope:** reactiveはまだ実際に使ったことはない Jan 25
ので、ちょっと比較できないです。
ただ、Arrowのレベルの抽象化だと、その時々の入力と出力について書くのはいいんだけど、「これをやって、あれをやって」みたいな高レベルの制御を書きにくいようには感じました。Reactiveだとその辺りどうなのかなあ。

  **Shelarcy and Ope:** ↑ あれっ、二人で一つのコメントを編集できるみたいです Jan 25
ね。

  **Me and Ope:** 出来ます、他人のコメントも編集し放題w Jan 25

  **IKEGAMI Daisuke and Ope:** Playback という機能があるのですが、この議論 Jan 25
が終わった後、試してみるといいです。一字一句いちいち録画していて、逆戻りしたり、リソースがもったいない。

  **Nobsun and Ope:** ああこれ面白そう。 Jan 25

  **Me and Ope:** そろそろ時間ですね。 Jan 25
というわけで、とりあえずはお疲れさまでした～

  **Nobsun and Ope:** 乙 Jan 25

  **Shelarcy and Ope:** お疲れさまでした～☆🎵 Jan 25

  **Maoe and Ope:** お疲れさまです Jan 25

  **Me and Ope:** 今回、初めて Google Wave を使ってみたわけだけど、結構面白いです Jan 25
ね、これ。
ただ重いなあ。

  **IKEGAMI Daisuke and Ope:** 昨日も使ったんですが、Mac でFirefox じゃない人はIME Jan 25
が途中でこけるという症状も見受けられました。今回は、全員喋れたようで何よりです、おつかれさまでした。



Shelarcy and Ope: IME ではないのですが、Safari で使っていて Reply がうまく Jan 25
できなくなったので、一回ログアウトしました。(ちなみに Opera だとそもそも Wave の
利用自体ができません.....。)



Me and Ope: ところで、今回のログを Google Wave のアカウントを持っていない人に Jan 25
も公開するにはどうやるのがいいのかなあ.....



IKEGAMI Daisuke and Ope: 各Waveの右下にFilesというプルダウンメニューがあるの Jan 25
だが、その "Download all" してもなにも起こらない Firefox



Me and Ope: それって、添付ファイルの操作のメニューだと思います。 Jan 25



IKEGAMI Daisuke and Ope: [http://www.googlewaveinfo.com/200910/how-to-print-from-
google-wave/](http://www.googlewaveinfo.com/200910/how-to-print-from-google-wave/) Jan 25

これを嫁と、いろいろぐぐった結果



Shelarcy and Ope: "Step 1 – Use [Firefox](#). Google Chrome works great for Wave, Jan 25
but Firefox is needed for this trick." へー



Me and Ope: おお、それでPDFにでもすればいいか。 Jan 25



IKEGAMI Daisuke and Ope: こちらは、画面に見えている部分しか PDF にならな Jan 25
い...ログを取るために Google Chrome インストールしたのに、やはり一ページだけ...



Me and Ope: むー、User Agent Switcher で User Agent を iPhone に切り替えたけど、 Jan 25
FirefoxでWaveのページが表示できない。



IKEGAMI Daisuke, Shelarcy and Ope: あと、うっかり[X]ボタン押して Wave 消 Jan 25
しちゃうと、もう全部消えるそうです。全世界の人々が怒っていて第三次世界大戦勃発前みた
いですよ...



IKEGAMI Daisuke and Ope: 明日の準備もあるので、諦めて寝ることにします...Wave Jan 25
さんがんばって...



Me and Ope: おやすみなさい Jan 25

Tags: [日本語](#) [haskell](#) [HIMA](#)

Next wave