石井 大海（いしい ひろみ）

筑波大学数理物質科学研究科
数学専攻 博士後期課程
E-Mail: h-ishii@math.tsukuba.ac.jp
Web Site: https://konn-san.com
GitHub: https://github.com/konn

基本情報

氏名　石井 大海（いしい・ひろみ）
性別　男
生年月日　1992 年 1 月 18 日

教育・学位

博士（理学），予定　筑波大学数理物質科学研究科数学専攻 博士後期課程
2016 年 4 月〜現在

修士（理学）　筑波大学数理物質科学研究科 博士前期課程・修了
2016 年 3 月　**修士論文**: On Regularity Properties of Sets of Reals and Inaccessible Cardinals

学士（理学）　早稲田大学基幹理工学部数学科・卒
2014 年 3 月

職歴・採用歴

2017 年 4 月〜2019 年 3 月　日本学術振興会特別研究員 DC2
　研究課題：『実数の集合の性質の集合論的解明と工学的応用』
2017 年 4 月〜　筑波大学数学専攻計算機管理オブザーバー
2014 年〜2017 年　筑波大学数学類『計算機演習』ティーチング・アシスタント
2014 年 4 月　Google Summer of Code 2014 採択
　題目：『Haskell による効率的な Gröbner 基底計算とそのための疎行列対角化アルゴリズムの実装』
2014 年 4 月〜2017 年 3 月　筑波大学数学専攻計算機管理アルバイト（www-admin）
2010 年 10 月〜2014 年 3 月　株式会社 Preferred Infrastructure アルバイト
2010 年 8 月〜9 月　株式会社 Preferred Infrastructure インターン生

賞罰・特記事項

- 日本学術振興会特別研究員 DC2,（2017 年〜2019 年）

- 第十四回茗渓会賞（2016 年）
- 2013 年度早稲田大学基幹理工学部長賞最優秀賞（第一回）
- 2013 年度早稲田大学基幹理工学部卒業生総代
- Web 数式処理ライブラリ KaTeX コミッタ
- Haskell 製 Web フレームワーク Yesod コミッタ

出版前レビュー担当書

- 『すごい Haskell 楽しく学ぼう！』，M. Lipovača ＝著，田中英行・村主崇行＝共訳，オーム社，2012 年，ISBN: 978-4274068850.
- 『型システム入門——プログラミング言語と型の理論——』，B. C. Pierce ＝著，住井英二郎＝監訳，遠藤侑介・酒井政裕・今井敬吾・黒木裕介・今井宜洋・才川隆文・今井健男=共訳，オーム社，2013 年，ISBN: 978-4274069116.
- 『Haskell による並列・並行プログラミング』，S. Marlow ＝著，山下伸夫・山本和彦・田中英行＝共訳，オライリー・ジャパン，2-14 年，ISBN: 978-4873116891.
- ほか 1 冊

研究業績

研究分野概説

主な研究分野は**数理論理学**，とくに**公理的集合論**と呼ばれる分野である．公理的集合論は主に実数の集合や，一般の無限について研究する数学の一分野である．集合論においてモデルの構成に用いられる**強制法**は，理論計算機科学における λ-計算のモデルである Scott **領域**と深い関係があり，Krivine の実現可能性モデル [3, 1, 2] として結実した．

私の興味分野はこうした集合論のモデルの構成法一般であり，そうしたモデルにおける実数の振る舞いである．現在メインで取り組んでいる物は飽和イデアルという物から得られる集合論のモデルである．一方で，上記の実現可能性モデルのような，計算論的背景を持つモデルにおける実数の集合性質を，実際に計算機上で実現することは出来るか？ という問題意識も持っており，これが現在採用中の日本学術振興会特別研究員の研究課題の一つにもなっている．

参考文献

[1] Jean-Louis Krivine, Realizability algebras ii : new models of ZF + DC, version 0, Logical methods in computer science 8 (1:10 Feb. 27, 2012), DOI: 10.2168/LMCS-8(1:10)2012, arXiv: 1007.0825 [math.LO].

[2] ———, Realizability algebras iii: some examples, version 0, Mathematical structures in computer science (May 2016), pp. 1–32, DOI: 10.1017/S0960129516000050, arXiv: 1210.5065 [cs.LO].

[3] ———, Realizability algebras: a program to well order $\mathbb{R}$, version 0, Logical methods in computer science 7 (3:2 Aug. 10, 2011), DOI: 10.2168/LMCS-7(3:2)2011, arXiv: 1005.2395 [cs.LO].

査読付き学会発表

March 2016, Freer Monads, More Extensible Effects. Programming and Programming Language Workshop (PPL) 2016, Okayama-prefecture, Japan.

査読無し学会発表

November 2017, Reflection Principle and construction of saturated ideals on $\mathcal{P}_{\omega_1}\lambda$. Workshop on Iterated Forcing Theory and Cardinal Invariants, Kyoto-prefecture, Japan.

論文

[1] Hiromi ISHII, On regularity properties of set of reals and inaccessible cardinals, MA thesis, Tsukuba University, 2016.

[2] Oleg Kiselyov and Hiromi ISHII, Freer monads, more extensible effects, Proceedings of the 2015 acm sigplan symposium on haskell, Haskell '15, Vancouver, BC, Canada: ACM, 2015, pp. 94–105, ISBN: 978-1-4503-3808-0, DOI: 10.1145/2804302.2804319, URL: http://doi.acm.org/10.1145/2804302.2804319.

上記の [2] は関数型言語において複数の副作用を合成する手法を提案したものであり，Oleg Kiselyov 氏との共同研究である．

スキルセットと開発実績

主に関数型プログラミング言語 Haskell を用いた汎用プログラムの開発を得意とする．特に，Haskell の特性を活かした並行・分散処理の記述や，型システムの機能を応用した安全なソフトウェアの開発に大きな関心がある．得意とする開発スキルセットを以下に挙げる：

- **型システム**や**性質ベーステスト**などの**形式手法**を用いた**信頼性の高いプログラム**の開発．
  - 定理証明系 Agda [10] を用いた（構成的）数学の形式化経験もあり．
- **数学的なアルゴリズム**，特に計算機代数の効率的な実装．
- **高度な分散・並行処理**を用いたプログラムの開発．
- Haskell を用いた**実践的な** Web **アプリケーション**の開発．
- パーザや構文木の操作を通じたメタプログラミングや文書の生成．

開発経験のある言語は，頻度順で次の通りである：

- Haskell
- JavaScript, CoffeeScript
- Ruby (〜1.9), Objective-C
- Agda

また，筑波大学の数学類[*1]の学生を対象とした演習『計算機演習』では，2014 年度より Haskell を学生に教えているが，この講義の立ち上げ・運営に参画し，ティーチング・アシスタントとしてレポート採点システムの開発や授業計画案の提言などを行っている．また，後述する計算機代数ライブラリを用いた，数学類生の卒

[*1] 通常の大学で学部の数学科に相当する．

業研究にティーチング・アシスタントとして参加し，Haskell による開発手法を生徒らに教育した．こうした活動を通して，個人としての開発能力だけでなく，プログラミング教育のスキルも日々磨いている．

以下，過去に従事した開発事例を幾つか紹介する．

レポート自動採点システム λeport

上述した「計算機演習」において，生徒の提出したレポート（Haskell プログラム）を自動採点するためのツールである．演習のティーティング・アシスタントにはそこまで Haskell に習熟していない担当者もおり，また単なる目視による採点では見落としがあり得る．そこで，関数型プログラミングにおける**性質ベーステスト**（Proeprty-based Testing）の方法論を応用し，レポートの答案の形式仕様を与え，生徒の答案がその仕様を満たすかを自動でチェックし，結果を報告する**自動採点システム** λeport を構築した．この大まかな構造を図 1 に示した．

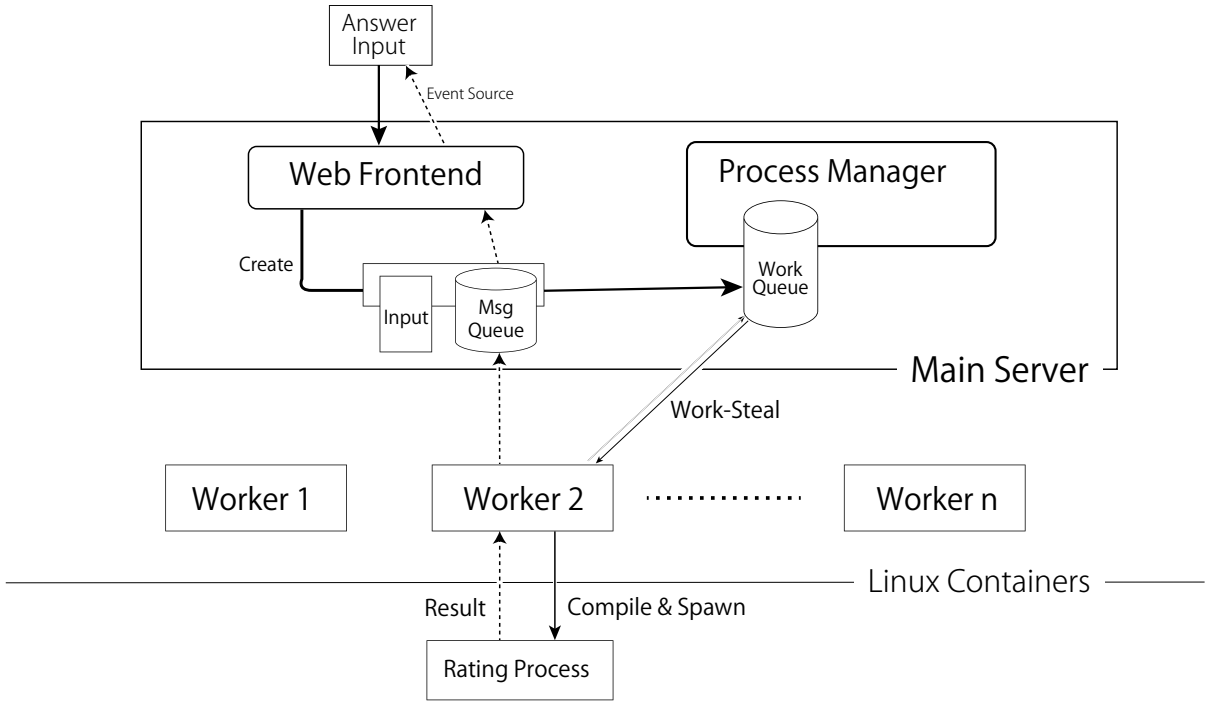

図 1　λeport の構造

λeport を設計するに当たっては，その性質上以下のような要請があった：

プライバシー　生徒の答案を扱うため，HTTPS 上で通信が行われる必要性がある．

並行性　複数人のティーティング・アシスタントが同時に採点を行うため，一度に複数の採点プロセスを走らせる必要性がある．

安全性　生徒の答案を入力するため，任意のコードが実行される畏れや，無限ループに陥りマシンリソースが食い潰される可能性がある．

リアルタイム性　採点結果はなるべく早く，各小問ごとに結果がわかるとよい．

以上の要請を満たすために，本システムでは以下の技術を用いている：

Let's Encrypt　**セキュア**な通信のために，Let's Encrypt [5] の発行する無料の証明書を利用し，通信を暗号化した．

Cloud Haskell　**並行性**を実現するため，Haskell による分散コンピューティングライブラリである Cloud Haskell [12] を採用した．図 1 に示した通り，問題の入力を受け付けたメイン・サーバのキューから別途ワーカ・プロセスが読み出す Work-Steal 方式により，効率的な分散処理を実現した．

Softoware Transactional Memory (STM)　一方，ワークキューへの登録や採点プロセスからのログのリレーなどメインサーバ内で並行処理をスレッド安全に実現するために，GHC の提供する Software Transactional Memory の機構を利用した．特に並行処理対応のキューを用いることで，ロジックを宣言的・受動的に書く事が出来，簡潔な記述が可能になった．

Linux Containers　Linux Containers [11] 技術を利用することで，サーバ上に隔離されたコンテナ環境を作り出し，生徒の答案をコンテナ内部でコンパイル・実行することで，**安全性**を確保した．Linux Containers には CPU やメモリの占有率を設定・制御する機構があるため，必要以上のリソースが採点プロセスによって食い潰される事も防ぐことができる．

実際には Linux Containers のコンテナマネージャである LXD インターフェースを介してコンテナの操作を行った．これは，Haskell の LXC バインディングでは使えない機能も多いのに対して，HTTP を介した REST API を提供している LXD の方が柔軟性や機能面で優位であったためである．LXD は WebSocket を経由したプロセス通信の枠組みを用意しており，これをリレーすることで**リアルタイム性**も実現した．

EventSource　採点プロセスと Web フロントエンドの**リアルタイム**通信には EventSource を用いた．類似技術の WebSocket に較べて，プロトコルやエンドポイントを切り替える必要がなく，通信はサーバからクライアントへの一方通行で良いため，EventSource を採用した．

QuickCheck　性質ベーステストライブラリである QuickCheck [2] を用いてレポート問題の形式仕様を記述し，ランダムテストを走らせることで答案の正誤を判定している．

Haskell 上の計算機代数システム `computational-algebra`

Web Site: `https://konn.github.io/computational-algebra`

関数型言語 Haskell の旗艦処理系である Glasgow Haskell Compiler（以下，GHC）が提供するリッチな型システムを活かし，高度な数学的計算を型安全に行えるようゼロから設計・実装した．具体的には，型システムや形式手法を応用することで以下の長所を実現した：

型安全　係数体や変数の数，順番などを型で表すことで，異なる多項式環の元を誤って足してしまう，といった誤操作を防止する．

高い表現力　型レベル自然数を用いることで，$\mathbb{F}_5$ や $\mathbb{F}_{25}$ といった有限体を扱うことが出来る．

直感的　上の設計により $\mathbb{Q}[x,y,z]$ と $\mathbb{Q}[x,z,y,w]$ は型上区別されるが，これらの間の埋め込み写像を型情報だけから自動的に計算出来る．また，多項式それ自体も本来の数式で書くのと近い形で `#x ^ 2 + #z * #x - 2` のように書くことが出来る．

正当性の保証　前述の性質ベーステストライブラリ QuickCheck [2] を用いてアルゴリズムの正当性を静的に

検証し，ライブラリの品質を保っている．

上述の Google Summer of Code 2014 においては，高速な Gröbner 基底計算アルゴリズムとして知られる F_4 [4] および F_5 [3] アルゴリズムの実装を試み，複数の行列ライブラリを統一的に扱うための枠組みや，代数計算向けのライブラリの整備などを進めた．その他にも，有理係数多項式の因数分解や，代数的数の計算機能なども実装されている．

特に，型安全性を最大限実現するために，GHC の型レベル自然数機能を強化する `type-natural` [8] パッケージや，コンパイル時に Presburger 算術ソルバによって型レベル自然数の扱いを改善するコンパイラプラグイン・`ghc-typelits-presburger` [6] を開発した．それらを用い，リストや配列，ベクトルなど任意のシーケンシャルなデータ型に対して，そこから固定長のコンテナ型を創り出せる `sized` [7] パッケージも合わせて開発した．

本ライブラリは，筑波大学数学類の卒業研究や，エクアドル大学のヤチャイ技術大学の研究プロジェクト等でも用いられている．

その他の開発活動

- Web 上で高速かつ高品質な数式描画を可能とする KaTeX [1] に幾つかのコマンド・環境を実装し，コミッタ権限を得た．これまで本格的な JavaScript 開発はしていなかったが，`npm` や `browserify`，`flow` などを用いたワークフローには半日程度でキャッチアップ出来た．
- 2014 年 3 月までの Preferred Infrastructure 社におけるバイトでは，Haskell を用いた汎用パーザ生成器や Twitter クローラの開発などに従事した．
- Haskell の Web フレームワーク Yesod の OAuth 認証ライブラリの開発（現在は引退）．
- 個人サイト `konn-san.com` のサーバはレンタル VPS 上で走らせている．当該サイトのコンテンツは静的サイトジェネレータ Hakyll [9] によって生成されているが，LaTeX のソースから HTML と PDF を同時に生成したいという個人的な要求から，LaTeX と HTML の間の構文変換器を自前で実装している．
- その他の開発プロジェクトは，GitHub 上（`https://github.com/konn`）にて見ることが出来る．

参考文献

[1] Khan Academy, Katex – the fastest math typesetting library for the web, 2018, URL: https://khan.github.io/KaTeX/.

[2] Koen Claessen, Björn Bringert, and Nick Smallbone, QuickCheck: automatic testing of Haskell programs, 2018, URL: https://hackage.haskell.org/package/QuickCheck.

[3] Jean Charles Faugère, A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5), Proceedings of the 2002 international symposium on symbolic and algebraic computation, ISSAC '02, Lille, France: ACM, 2002, pp. 75–83, ISBN: 1-58113-484-3, DOI: 10.1145/780506.780516, URL: http://doi.acm.org/10.1145/780506.780516.

[4] Jean-Charles Faugère, A new efficient algorithm for computing Gröbner bases (F_4), Journal of pure and applied algebra 139.**1–3** (1999), pp. 61–88, ISSN: 0022-4049, DOI: http://dx.doi.org/10.1016/S0022-4049(99)00005-5, URL: http://www.sciencedirect.com/science/article/pii/S0022404999000055.

[5] Internet Security Research Group, ed., Let's encrypt, 2017, URL: https://letsencrypt.org.

[6] Hiromi ISHII, The ghc-typelits-presburger package, 2017, URL: http://hackage.haskell.org/package/ghc-typelits-presburger.

[7] ———, The sized package, 2017, URL: http://hackage.haskell.org/package/sized.

[8] ———, The type-natural package, 2013, URL: http://hackage.haskell.org/package/type-natural.

[9] Jasper Van der Jeugt, Hakyll - home, 2018, URL: https://jaspervdj.be/hakyll/.

[10] Agda Team, ed., The agda wiki, 2018, URL: http://wiki.portal.chalmers.se/agda/pmwiki.php.

[11] Linux Containers Team, ed., Linux containers, 2017, URL: https://linuxcontainers.org/ja/.

[12] Well-Typed, CloudHaskell Team, and Tim Watson, Cloud haskell, 2017, URL: http://haskell-distributed.github.io.