会議レポート

ICFP 2015 参加報告
―関数型プログラミングの今，来年の日本開催に向けて―

ICFP とは

2015 年 8 月 30 日から 9 月 5 日まで，関数型プログラミングに関する国際会議 ICFP（International Conference on Functional Programming）およびその併催ワークショップがカナダのバンクーバーで開催された．前身の「LISP および関数型プログラミング」（Lisp and Functional Programming）と「関数型プログラミングと計算機アーキテクチャ」（Functional Programming and Computer Architecture）が合併してから，20 回目の開催となる．メインカンファレンスの前後には，12 のワークショップと 13 のチュートリアルが開催された．各プログラミング言語に関するもの（Haskell・ML / OCaml・Erlang・Scheme）とさまざまなテーマ（副作用の理論・ジェネリックプログラミング・関数型高性能コンピューティング・関数型芸術と音楽・関数型プログラミングの商用利用）のものに加えて，今回は新しく学生にさまざまな知識を伝えることを目的とする「プログラミング言語指導ワークショップ」（Programming Language Mentoring Workshop）も開催された．

ワークショップ（本会議前）

残念ながら筆者が上記のすべてのワークショップに出席することはできなかったが，見たものの中で印象に残った発表について語っていきたい．

1 日目は前述の指導ワークショップに加えて HIW（Haskell Implementors Workshop），HOPE（Workshop in Higher-order Programming with Effects）と WGP（Workshop on Generic Programming）が開催された．筆者は HIW と HOPE の間を行き来した．HIW は GHC コンパイラの指揮をとる Simon Peyton-Jones の基調講演から始まった．その中で，来年発表される GHC 8 の新しい機能の紹介があり，型システムの強化で，Haskell への依存型（型の中で値の計算や性質を表現する能力）の導入が着々と進んでいるのが印象的だった．また，Haskell の古い問題である，異なるレコード型で同じフィールド名が使えないことについて，型クラスのさまざまな拡張を利用して解決するという話も型技術者の魂をよく表現していた．

図 -1　Rastislav Bodik の招待講演

その次に移動した HOPE では，マイクロソフトの研究所で開発されているプログラミング言語 F*（エフスター）の紹介が興味深かった．依存型と自動証明の導入により，プログラムのさまざまな性質の自動検証が可能になり，安全なプログラミングがもはや夢ではなくなっていることを物語っていた．

本会議

2 日目から ICFP 本会議が始まったが，いきなり立見席が出るというハプニングがあった．今回の ICFP は本会議のみでも参加者が 350 人を超えており，用意された部屋の収容人数を超えていた．その日の午後から席を増やしたりして問題が解決されたが，最近の ICFP が安定して多くの参加者を集めていることの表れでもある．

2 日目の招待講演であった Rastislav Bodik の「Program Synthesis : Opportunities for the Next Decade」が今回の ICFP の 1 つのテーマを与えた．Bodik 氏自身は関数型プログラミングコミュニティの人間ではないが，彼が語ったプログラムの自動生成や網羅的な最適化の話が，今の関数型プログラミングの行くべき方向の 1 つである．一般発表でも，プログラム生成と関連のある発表が少なくなかった．プログラムの生成に関数型プログラミング言語が利用されることも多いが，むしろ今後興味があるのは，そこで使われる型システムがどうプログラム生成を効率化できるかと，関数型プログラミングの証明テクニックの応用だろう．

次のコンパイラのセッションでの Adam Chlipala による「An Optimizing Compiler for a Purely Functional Web-Application Language」はその一例だろう．型システムとプログラム変換を利用した徹底的なプログラム最適化

を使って，最速の Web フレームワークを開発した話である．

プログラム委員長の John Reppy が行った委員会報告には，研究を推進する会議としての健全な姿があった．全世界（5 大陸）から 119 件の投稿があり，その中から 35 件が選ばれた．倍率は 3.3 倍とそれほど高くないが，そのお陰でプログラム委員会が良いと思った論文を素直に選べるというありがたみがある．パラレル・セッションを使わずに，すべての発表を全員が聴けるのも，興味が近いコミュニティとしてありがたい．

3 日目に行われた招待講演は Mary Sheeran の「Functional Programming and Hardware Design」だった．あまり知られていない，もう 1 つの関数型プログラミングの応用分野の歴史を通じて，技術的な成功を認めてもらうことの難しさが窺えた．プログラム生成との関連もあり，前日の招待講演とうまく呼応していた．

その次のセッションのテーマは定理証明であり，このコミュニティで形式的な証明が日常的に行われていることを再認識できた．

その日の夕方には 10 年前の ICFP 2005 で最も影響を与えた論文の発表があった．選ばれたのは Manuel Chakravarty らの「Associated Type Synonyms」．要素の型によってコレクションの表現を変更するために，Haskell に初めて型レベルの関数を導入した研究であり，それが大きな流れを作った．

それに続いて，ICFP プログラミング・コンテストの結果発表も行われた．このコンテストでは，あるタスクを 3 日間で実装しなければならない．使う言語は自由で，今年も 300 以上のチームが参加し，正六角の盤面上でのテトリスのようなゲームに励んだ．結果はなんと，1 位と 2 位は日本のチームだった．複数の言語を使うチームが多く，2 年ぶりに優勝した Team Unagi の場合では C++・PHP・Haskell などが使われた．

4 日目には，招待講演の代わりに，今年亡くなられた Paul Hudak の追悼セッションがあった．関数型プログラミングに多くの貢献をした同氏だが，理論的なものと同時に，グラフィクスとコンピュータ音楽での貢献も思い出される．Conal Elliott と一緒に Functional Reactive Programming を提案し，絵や動画を純粋に数学的な方法で表現する方法を与えた．晩年，関数型プログラミングと芸術や音楽の関係を追求する FARM ワークショップの立ち上げのためにも働いた．

その日にはコントラクトのセッションもあったが，ワークショップを含めて多数の発表があった．コントラクトとは，アサーションが失敗したとき，原因が呼び出し元か呼び出し先のどちらにあるかを決めるための技術で，高階関数のある関数型プログラミングでは有用な機能である．関数型プログラミング言語の多くは強い型システムを持っているものの，型だけで表現できない不変量をコントラクトという形でプログラムに埋め込んで，正しく責任を割り当てるのが今でも活発に研究されている．

図 -2　Paul Hudak の追悼セッション

ワークショップ（本会議後）

本会議の後の 3 日間はワークショップとチュートリアルの期間である．伝統的には，本会議前には理論的なワークショップを配置し，本会議後にはより実装に近いテーマを扱う．その理由の 1 つは最終日に CUFP（Commercial Users of Functional Programming）が開催され，企業から多くの参加者が集まることである．本会議が終わってからワークショップだけのために来る人も少なくない．

2 日連続で開催される Haskell Symposium に人が最も多く集まるが，筆者は ML Family Workshop と OCaml Workshop に参加した．OCaml については，opam という独自のパッケージシステムがますますさかんで，コミュニティの広がりから，各パッケージの開発形態の研究にも使えるという面白い発表もあった．

このようなワークショップとは別に，さまざまなシステムに関するチュートリアルも行われた．

次は奈良で

7 日間にわたって開催された ICFP 2015 だが，面白い発表をたくさん聴いて，参加者全員が満足して帰ったものと思う．来られなかった人のために，ほとんどの発表を YouTube で見ることもできる[☆1]．そして，本会議の最後に紹介された ICFP 2016 の会場は奈良なので，来年は日本からたくさんの投稿と多くの参加者を期待したい．

（Jacques Garrigue ／名古屋大学多元数理科学研究科）

☆1　http://icfpconference.org/icfp2015/