

令和元年6月3日現在

機関番号：11301

研究種目：若手研究(B)

研究期間：2015～2018

課題番号：15K15966

研究課題名(和文) 双方向変換記述のための高階関数プログラミング言語

研究課題名(英文) A Higher-Order Programming Language for Bidirectional Transformations

研究代表者

松田 一孝 (Matsuda, Kazutaka)

東北大学・情報科学研究科・准教授

研究者番号：10583627

交付決定額(研究期間全体)：(直接経費) 2,500,000円

研究成果の概要(和文)：双方向変換は、複数のデータ間の同期を可能とし、プログラミング言語、ソフトウェア工学、データベース等様々な分野で注目を集めている重要な研究課題である。

本プロジェクトでは、双方向変換記述のための高階関数プログラミング言語の設計を行った。この言語では、双方向変換は通常の関数として表現されるため、ユーザはその関数表現を高階関数を用いて組み合わせることにより複雑な双方向変換を記述することが可能となっている。また、束縛子を持つような双方向変換プログラミング言語の新しい埋め込み実装法を提案し、その有用性を構文解析器/整形出力器記述のための双方向言語FliPprを埋めこむことで確認した。

研究成果の学術的意義や社会的意義

プログラミング言語の記述性は、ソフトウェアの生産性や保守性に関係する重要な性質の一つであり、双方向プログラミング言語においても例外ではない。本研究の提案言語は双方向変換言語の記述性を大きく向上させるものである。本研究で提案された言語のアイデアは、双方向変換を通常の関数として表現することである。このことは双方向変換をより自然なプログラムとして表現することを可能にするだけでなく、高階関数を用いた双方向プログラミングをサポートすることを可能にしている。また、埋め込み技術の発展は双方向プログラミング言語の利便性の向上につながる。

研究成果の概要(英文)：Bidirectional transformation, which enables synchronization between multiple data, is an important research topic that has been attracting interest in various research disciplines, including programming languages, software engineering, and database systems. In this research project, we developed a higher-order functional programming language for bidirectional transformations, in which bidirectional transformations are represented as ordinary functions so that users can compose them by using higher-order functions to develop involved bidirectional transformations. We also proposed a new method for embedding bidirectional programming languages with binders, and demonstrated its usefulness by a case of FliPpr, a bidirectional programming language that describes parsers and pretty-printers at the same time.

研究分野：プログラミング言語

キーワード：プログラミング言語 関数プログラミング プログラム変換 領域特化言語 双方向変換

様式 C-19、F-19-1、Z-19、CK-19 (共通)

1. 研究開始当初の背景

双方向変換は、ソースと呼ばれるデータからビューと呼ばれるデータを得るための通常の順方向の変換と、ビュー上の変更をソースの上へと反映させる逆方向の変換の二つの変換からなる。双方向変換は、データベースにおけるビュー更新問題のみならず、異なる形式のデータの同期、GUIプログラミング、Webアプリケーション作成、アンドウ/リドゥ機構の実現、整形出力プログラムと構文解析プログラムの作成、およびソフトウェアのモデル駆動開発におけるモデル間の同期等、様々な応用を持つ。そのため**高信頼の双方向変換を効率的に作成できれば、ソフトウェアの生産性および信頼性の向上につながる**ことが期待される。双方向変換は、プログラミング言語・データベース・ソフトウェア工学の分野で学際的に研究されており、双方向変換を専門的に扱った国際ワークショップも 2012 年より毎年開催されているように近年盛り上がりを見せている。

双方向変換は双方向に動作すること、つまり「双方向性」を満たすことが要求される。たとえば、ビューを何も変更しなければソースは何も変更されないことが望ましく、またビューを更新した場合には、その更新を実現されるようにソースを対応して変更されることが望ましい。

双方向変換プログラミング言語には課題がまだ残されている。ひとつの課題は、**双方向変換プログラミング言語の記述性**である。理想的には、双方向変換プログラムをできるかぎり順方向のプログラムと同程度の手間で記述したい。しかしながら、既存の双方向変換プログラミング言語は通常のプログラミング言語とギャップが大きく、特に**双方向変換プログラミング言語では高階関数を自由に利用できない**。このことは、関数プログラミング技術を双方向変換プログラミングに適用することを妨げ、その結果、双方向変換プログラム作成の手間が通常のプログラミングと比べて大きい原因の一つとなっている。

一方で**双方向変換プログラミング言語の表現力**も問題となる。順方向の変換に対し、「双方向性」を見出すような逆方向変換は一般に複数あるため、双方向プログラミング言語は、その違いを適切プログラミングする機能を有することが望ましい。

2. 研究の目的

本研究は、**高階関数プログラミング技術をそのまま応用可能な、より生産性/利便性の高い高階双方向変換プログラミング言語の設計・実装**を目指す。

3. 研究の方法

双方向プログラミングにおける一つのアプローチは、最も有名な双方向プログラミング言語である lens 等に代表されるような**コンビネータに基づくプログラミング**である。この方法では、双方向コンビネータを用いて小さな双方向変換を合成することで大きな双方向変換を作成する。そのため、問題に合わせて適切に双方向プリミティブやコンビネータを選ぶことで様々な双方向変換が表現可能である一方、ユーザは特殊なプログラミングスタイルを強いられるため記述性は低い。また別のアプローチは、Voigtländer や研究代表者らによって研究されてきた、通常の一方向のプログラムから双方向プログラムを導出する**双方向化**である。この手法では、ユーザは高階関数を含む通常のプログラムを記述できる一方で、一方向のプログラムのみを記述するため双方向変換としての表現力は限定的である。研究代表者は、この二つの**アプローチを融合**させることにより、高階双方向プログラミングの実現を目指した。

また、言語の生産性/利便性を高めるために、双方向プログラミング言語の埋め込み実装技術についても研究を進めた。埋め込みプログラミング言語とは、Haskell や OCaml などの別の言語（ホスト言語）のライブラリの形で実装されるプログラミング言語であり、ホスト言語の型システムや、処理系や IDE 等のエコシステムをゲスト言語が利用可能であるという利点がある。

4. 研究成果

本研究の成果は以下の通りにまとめられる。

(1) **高階双方向変換プログラミング言語** 本研究の一つの大きな成果は高階双方向プログラミング言語の実現である。提案言語では、双方向変換は通常の高階関数として表現される。そのため、ユーザは双方向変換を通常の方の関数プログラムに近い形で実装できるだけでなく、高階関数を用いて双方向変換を組み合わせることでより複雑な双方向変換を記述することが可能である。本研究の技術的な面白さの一つは、これまでに研究された二つの双方向プログラミング技法、コンビネータに基づく双方向プログラミングと、プログラムの双方向化のそれぞれ利点を組み合わせることを可能とした点である。

この成果の一部は、この分野のトップレベル会議である ICFP および ESOP に採録されている。また、ICFP で発表した内容を整理した論文がこの分野のリーディングジャーナルの一つである Journal of Functional Programming に採録された。

(2) **束縛子を持つ双方向変換プログラミング言語の埋め込み手法** 前述の通り言語の埋め込み実装は、ゲスト言語にホスト言語の機能やエコシステムを利用可能にし、利便性を向上させる。縛子 (パターンマッチや関数定義などの変数を導入するような言語要素) を持つようなプログラミング言語の有用な埋め込み法の一つは、PHOAS (parametric higher-order abstract syntax) や finally-tagless style などの高階抽象構文を用いるものである。しかし、束縛子を持つような双方向変換プログラミング言語は高階抽象構文を用いた埋め込みが自明ではなかった。このことは、双方向変換プログラミング言語における「関数」が、数学的な関数に対応しないことによる。研究代表者らは、Robert Atkey によって提案された unembedding 技法が束縛子を持つような双方向言語の埋め込みに有用であることを発見し、その有用性を双方向言語の一つ FliPpr の Haskell に埋め込むことで確認した。FliPpr は研究代表者らの過去の研究成果の一つである。FliPpr では、ユーザが制限された構文にて構文木の整形出力器を書くことにより、同時にその整形出力器に対応する構文解析器を得ることができる。

この成果の一部は、Haskell Symposium に採録された。また、ここで言及した FliPpr については、過去の研究を整理した内容がジャーナル New Generation Computing に採録された。

(3) **グラフ変換言語 UnCAL の再整理** グラフ変換言語 UnCAL は Buneman らによって提案されたグラフ変換言語であり、その関数プログラミング技術との相性のよさから NII のグループによってグラフ間の双方向変換の研究にも利用されている。一方で、UnCAL は様々なグラフ変換に特化した言語要素を含み、そのことが関数プログラミング技術の直接の適用を妨げてきた。研究代表者らは、UnCAL の変換がそれらのグラフ変換に特化した言語要素を用いることなしに、実現可能であることを示した。

この成果の一部はプログラム変換等に関する国際ワークショップ PEPM に採録された。また、その成果のさらなる一般化である、UnCAL 変換の代数的な定式化に関する論文がジャーナル Mathematical Structures in Computer Science に採録されている。

5. 主な発表論文等

[雑誌論文] (計 8 件)

以下は全て査読有り。

- [1] [Kazutaka Matsuda](#), Meng Wang: Applicative bidirectional programming: Mixing lenses and semantic bidirectionalization. *Journal of Functional Programming* 28: e15 (2018)
- [2] Makoto Hamana, [Kazutaka Matsuda](#), Kazuyuki Asada: The algebra of recursive graph transformation language UnCAL: complete axiomatisation and iteration categorical semantics. *Mathematical Structures in Computer Science* 28(2): 287-337 (2018)
- [3] [Kazutaka Matsuda](#), Meng Wang: FliPpr: A System for Deriving Parsers from Pretty-Printers. *New Generation Computing* 36(3): 173-202 (2018)
- [4] [Kazutaka Matsuda](#), Meng Wang: HOBiT: Programming Lenses Without Using Lens Combinators. *ESOP 2018*: 31-59
- [5] [Kazutaka Matsuda](#), Meng Wang: Embedding invertible languages with binders: a case of the FliPpr language. *Haskell@ICFP 2018*: 158-171
- [6] [Kazutaka Matsuda](#), Kazuyuki Asada: A functional reformulation of UnCAL graph transformations: or, graph transformation as graph reduction. *PEPM 2017*: 71-82
- [7] [Kazutaka Matsuda](#), Meng Wang: Applicative bidirectional programming with lenses. *ICFP 2015*: 62-74
- [8] Soichiro Hidaka, Martin Billes, Quang Minh Tran, [Kazutaka Matsuda](#): Trace-based Approach to Editability and Correspondence Analysis for Bidirectional Graph Transformations. *Bx@STAF 2015*: 51-65

[学会発表] (計 5 件)

- [1] 菅野 翔太, [松田 一孝](#), Oleg Kiselyov: Linear Quipper: 埋め込み線形型付き量子プログラミング言語。第 21 回プログラミングおよびプログラミング言語ワークショップ, 3 月, 2019. 査読有り。
- [2] 八木 颯, [松田 一孝](#): 負型と分数型を持つ線形型付き可逆プログラミング言語とその並行計算に基づく意味論。日本ソフトウェア科学会第 35 回大会, 9 月, 2018. ポスター
- [3] 菅野 翔太, [松田 一孝](#): Linear Quipper: 埋め込み線形型付き量子プログラミング言語。日本ソフトウェア科学会第 35 回大会. 9 月, 2018
- [4] [松田 一孝](#), Meng Wang: HOBiT: A Higher-Order Language that Bridges Uni- and Bi-directional Programming. 日本ソフトウェア科学会第 33 回大会, 2016. ポスター

- [5] Kazutaka Matsuda and Meng Wang: *Applicative Bidirectional Programming with Lenses*.
日本ソフトウェア科学会第32回大会, トップカンファレンス特別講演. 9月, 2015.

[その他]

研究代表者 Web サイト : <http://www2.sf.ecei.tohoku.ac.jp/~kztk/>
HOBiT の実装 : <https://bitbucket.org/kztk/hibx/>
埋め込み版の FlipPr の実装 : <https://bitbucket.org/kztk/flippre/>
Applicative Lens の実装 : <https://bitbucket.org/kztk/app-lens/>

6. 研究組織

(2) 研究協力者

研究協力者氏名 : WANG, Meng

※科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属されます。