様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成　２７　年　　６　月　１６　日現在

機関番号：　３２６８９
研究種目：　挑戦的萌芽研究
研究期間：　2012～2014
課題番号：　２４６５００１６

研究課題名（和文）検証技術と非標準型体系を用いたモデル検査器コンパイラの進化的発展

研究課題名（英文）Evolutionary development of a model checker compiler using verification technology and non-standard type systems

研究代表者

上田　和紀（Ueda, Kazunori）

早稲田大学・理工学術院・教授

研究者番号：１０２５７２０６

交付決定額（研究期間全体）：（直接経費）　　2,900,000円

研究成果の概要（和文）：非手続き型高水準言語の適用分野として，さまざまな系のモデリングと検証が今後重要性を増すと期待される．研究代表者らはモデル検査機能を通常の計算と同一の枠組で提供するグラフ書換え言語処理系の開発と公開を進めてきたが，コンパイラの進化的発展と信頼性確保が課題であった．本研究では，コンパイラと実行時処理系をつなぐ抽象機械によるグラフ書換え操作の形式化と検証を行うことで，コンパイラの検証に向けた一歩を踏み出した．また，モデルの解析，理解，最適化に役立つグラフ書換え言語のための新たな型体系を，微視的な接続関係に着目した体系とグラフ全体の形状を扱う体系の両面から検討してそれぞれ構築した．

研究成果の概要（英文）：Modeling and verification of various systems are expected to be important applications of non-procedural high-level languages. The Principal Investigator has long worked on the design and implementation of a modeling language based on graph rewriting that supports ordinary execution and model checking in a unified framework, but the evolution and verification of its compiler has been recognized as a challenging issue. In this study, we formalized and verified graph rewriting operations performed by the abstract machine that is the interface between the compiler and the runtime, and made a step towards the verification of a graph rewriting compiler. Also, we established novel type systems for the graph rewriting language that will help the analysis, understanding and compiler optimization of models. One of them handles microscopic connectivity of graphs and hypergraphs, while the other handles the global shape of graphs.

研究分野：　プログラミング言語

キーワード：　モデル検査　コンパイラ　検証　型体系　ソフトウェア進化

１．研究開始当初の背景

研究代表者は，並行性と制約概念をもつ非手続き型プログラミング言語の研究を，言語設計，理論，処理系開発の三面から長期にわたり推進してきた．手続き型言語が非手続き型言語の概念を徐々に吸収して進化しつつある中，非手続き型の高水準言語の今後の適用分野として，さまざまな系のモデリングと検証が重要性を増すと期待され，近年は，当初階層グラフ書換えに基づく計算モデルとして提案した LMNtal 言語を，モデル検査機能を通常の計算と同一の枠組で提供する高水準モデリング言語処理系へと発展させ，約 8 年の年月をかけて 10 万行を超す規模のシステムを構築してきた．

LMNtal モデル検査器は抽象機械へのコンパイラ（記述言語は Java）と抽象機械コードの実行系（記述言語はC）からなる．実行系に関してはプロトタイプ開発時点からさまざまな改良拡張が行われ順調に発展してきたが，コンパイラは，通常の言語処理系からモデル検査器へと発展する以前からのものがそのまま使われていた．LMNtal のようなグラフ書換え言語の抽象機械コード生成は高度な変換作業であって，コンパイラの発展改良は実行時処理系の発展改良よりも難度が高いというのが開発チームメンバの共通認識であった．

２．研究の目的

本研究では，非手続き型高水準モデリング言語コンパイラの信頼性と保守性を今後抜本的に改善してゆくための道筋として，(a) 証明支援技術を用いた進化的コンパイラ検証，および (b) 非標準型体系の導入とそれによるコンパイラ発展，の二つを柱とした研究を行い，コンパイラを進化的に世代交代させてゆくことを目標とする．対象とするコンパイラは約28,000行の規模であり，コード生成部はモノリシックな構成をとっている．

本研究は，(i) 非手続き型高水準言語のコンパイラを扱う点，(ii) コンパイラ構成法が未成熟な分野を扱う点，(iii) 検証作業がコンパイラの進化を後押しする点，(iv) 実用規模のコンパイラを扱う点，(v) 非標準型体系の活用によってコンパイラ進化を図る点などが挑戦的であり，当初の計画通りに進むかどうかは明らかではないが，実用規模の言語と処理系に適用可能であることを担保しつつ，要素技術の開拓を進めてゆく．

３．研究の方法

モデリング言語 LMNtal は，高水準言語プログラムが直接モデル検査できる処理系をもつことを特徴とするが，グラフ書換え言語プログラムの抽象機械コードへのコンパイルは大きなセマンティックギャップの橋渡し作業であり，その信頼性を確保する方法は自明ではない．そこで，目的の達成に向けて以下の各側面からアプローチすることにした．

まずコンパイラが生成する抽象機械コードの正当性検証を可能にして，さらに将来の発展を可能にするには，抽象機械の設計を再検討してその抽象度を適切なレベルに設定する必要がある．そこで，過去のコンパイラ保守改良の経験に基づいて抽象機械の命令体系および生成されるコードを再検討しつつ，その正当性を順次検証してゆくこととした．

次に非標準型体系については，過去に提案されたグラフ書換え言語の型体系および並行制約言語の非標準型体系（K. Ueda, TACS'01, 95-126, 2001）の研究を参考にしつつ，ハイパーリンク（ハイパーエッジ）を扱うことのできる型体系を確立させてその有効性を確認することとした．またこれと並行して，上記の型体系がカバーしないグラフの大域的側面を扱う型体系の検討を行うこととした．

上記の基礎検討やその試験実装から知見を得ながら，稼働中のLMNtalコンパイラの保守改良を進め，技術およびスケジュールの観点から可能な範囲で，新たな知見を処理系に反映させてゆくこととした．

４．研究成果

(1) LMNtal 抽象機械の再設計と改良

LMNtal プログラムは，アトムと呼ばれる頂点とリンクと呼ばれる辺から成るグラフの書換えルールの集合である（本報告書では LMNtal のもう一つの重要概念である膜については省く）．LMNtal の抽象機械命令列はレジスタ変数を用いた SSA（静的単一代入）形式であり，各書換えルールの左辺のマッチング処理と右辺への書換え処理を手続き的に実現する．マッチングを処理する命令列をヘッド命令列，マッチした構造に対する書換えを行う命令列をボディ命令列と呼ぶ．

ボディ命令列では，アトムの作成(newatom)と削除(removeatom)，アトム間相互接続に用いる 3 種類の基本的なリンク操作命令(newlink, unify, relink)，および最適化コンパイル時に用いる 2 種類のリンク操作命令(getlink, inheritlink)を用いてグラフの書換えを実現していた．しかし，LMNtal コンパイラの備える最適化器によって変換された命令列が，特定の条件下において正しくないグラフ書換えを実行してしまうという問題が，解決法が見出せないまま長らく存在していた．

抽象機械命令列が元のグラフ書換えを正しく実現することを形式的に検証することでコンパイラの正当性を保証する必要性が高まったが，現状の LMNtal 処理系をそのままの形で検証することは，次の点から困難だと判断した．まず，ある種の最適化（具体的には，グラフ書換えにおける書換え前のアトムの再利用）を施す場合に既存の命令体系ではそもそも実現できない操作が存在する．さ

らに，その対策として導入されたいくつかの命令は，グラフ構造よりも抽象度の低い参照概念を直接扱うものであり，参照のエイリアシング（複数の参照が同一のアトム引数を指す）などの観点から命令列の形式的な扱いが難しくなる．

そこで，relink 命令の代わりとして接続先の互換を行う swaplink 命令を新規に設け，グラフ書換え操作の手順を代数的操作に帰着させることに成功した．swaplink 命令は，well-formed なグラフの 2 本のリンクの接続相手を交換するものである．これに基づくグラフ書換え操作の定式化は，既存の最適化手法の効果を損なうことなく従来の問題点を解決し，かつ形式化に向いた数学的性質を持つことが判明した．

(2) グラフ書換え操作の Coq による形式化

グラフ構造は，その表現力の高さからモデルの記述能力に優れている．しかし，グラフ構造の表現と操作の形式化は，リストや木のような再帰的データ構造と違って自明ではない．グラフの数学的定義になるべく忠実な形をとるのか，プログラム中での表現に忠実な形をとるのか，などの選択肢がある．さらにグラフは，ループや多重辺の有無，辺の向きの有無などで多くのクラスに分けることができ，成り立つ性質や適切な定式化の方法もそれぞれ異なる．このため，一般的なグラフに対する証明支援系のライブラリは存在するものの（Coq の GraphBasics 等），再利用性は高いとはいえない．

本研究では，グラフ構造の定義として，頂点集合と辺集合を用いた数学的表現ではなく実装に近い構造を選択した．この選択によってグラフ構造の扱いは多少複雑になるが，グラフ書換えをその実装に近い抽象度で厳密に論じることができる．グラフ書換え命令は，既存の抽象機械命令から選んだ newatom，removeatom，newlink，unify に swaplink を加えた 5 つを対象とし，その動作を Coq 上で定義した．

形式化した定義の上で成り立ついくつかの重要な性質の証明を行った．具体的には，形式化した各グラフ書換え命令の形式的仕様および LMNtal グラフの well-formedness 保存定理である．命令の形式的仕様とは，命令の実行前後におけるグラフ構造の条件であり，各命令の形式的意味論とみなすものである．LMNtal グラフの well-formedness とは，生成されるグラフ構造の LMNtal グラフとしての整合性である．本論文におけるグラフ構造の定義は，LMNtal グラフとみなせない不正な構造を許している．したがって，各命令における well-formedness の保存は命令列の実行の安全性を論じる上で重要な性質である．

保存定理の Coq 上での形式化は，以下の手順で行った：(a) 扱う対象となるデータ構造の定義，(b) データ構造に対する処理（関数）の定義，(c) 命題の定義，(d) 補題，定理の証明．(a)では，アトムやリンクの表現，それらからなるグラフ構造の表現を定めた．(b)では，実際の処理となるリンク書換え操作や，それを実現するための補助関数を実装した．(c)では，データ構造の間の関係や証明すべき性質を論理式で定義した．(d)では，補助関数の性質などの小さな補題の証明，およびそれらを利用したより大きな定理の証明を行った．

記述した証明スクリプトは全体で約 4100 行となった．グラフ書換えのために定義した関数は 17 個である．Well-formedness 保存定理を証明するために，221 個の補題を証明した．つまり Coq スクリプトの多くを占めているのは定理証明に関する記述である．このことから分かるように，Coq 上で定義した関数について性質証明を行うためには，その数倍の規模の証明を記述する必要があり，今回の検証実験によって LMNtal コンパイラ全体の検証に関する予備的知見を得ることができた．

(3) 実数領域を用いた非標準型体系

プログラミング言語とその実装における型体系の役割は広く認識されているが，グラフ書換えプログラムの解析や理解，およびグラフ書換えプログラムの実行やモデル検査の空間効率や時間効率の改善においても型体系が果たす役割は大きいと期待される．LMNtal については型体系の提案が過去にあったものの，処理系との連携はとられてこなかった．

また研究代表者らは，2 点間接続のリンクに加えて多点間を接続するハイパーリンクの有用性に着目して，LMNtal にハイパーリンクを導入した拡張言語 HyperLMNtal の仕様を定め処理系を構築してきたが，HyperLMNtal のための型体系はこれまで整備されてこなかった．

HyperLMNtal は，いかなる種類の宣言（変数宣言，型宣言，関数宣言等）もプログラマに要求しない簡潔なモデリング言語を目指しているが，その解析手法の開拓は，プログラムの理解や実行効率改善等の意義に加えて，言語の重要なサブクラスの同定にも役立つ．そこで，研究代表者の並行論理プログラムの非標準型体系に関する過去の研究を出発点として，HyperLMNtal の型体系を整備することとした．

グラフおよびグラフ書換えのための型体系としては，巨視的な，すなわちグラフ全体の構造を捉えるものと，微視的な，すなわち個々のアトム（グラフ頂点）の他との接続関係や個々のリンクの役割を捉えるものとが考えられるが，本研究では微視的なアプローチをとり，アトムの各ポート（アトムと引数番号で指定されるリンク端点）に対して，正負の極性の概念を一般化した属性値を付与

することとした．具体的には，実数領域 [-1,+1] で表現されるケイパビリティの概念を用いて各ポートの属性を表すこととした．値 +1 はアトムの当該ポートがその相棒アトム（リンクでつながったアトム）の排他的つまり唯一のオーナー（参照者）であることを表し，値 0 < c < 1 は非排他的つまり部分的オーナーであることを表す．値 -1 はデータの唯一のアクセスポイント（提供者）を表す．あるハイパーリンクが一つの -1 ポートといくつかの正小数ポートをつなぐとき，それは後者から前者への有向ハイパーリンクと解釈できる．値 -1 < c < 0 のポートは部分提供者を表すが，これは部分的オーナーシップを返却するときなどに現れる．値 0 はどこにも接続されていない非活性なポートを表す．

このようなケイパビリティに対して，グラフや書換えルールの構文から導かれる 4 種類の値制約を課することによって新たな型体系を構築した．つまりグラフや書換えルールが正しく型付けされているとはどういうことかを形式的に定義した．たとえば，個々のハイパーリンクは上述のようにちょうど 1 個の -1 ポートといくつかの正小数ポートをもつが，それらのケイパビリティの総和は 0 でなければならない．これは回路理論におけるキルヒホッフの電流則に相当する．他の 3 つの制約は，リンクやハイパーリンクどうしの相互接続における制約，各リンクやハイパーリンクがちょうど 1 個の負ケイパビリティを持たなければならないという制約，およびハイパーリンク以外のリンクのケイパビリティは整数（+1 または -1）でなければならないという制約である．

実数値制約を用いた代数的な定式化によって，型体系の基本定理，すなわちプログラムの実行が型を保存するという主部簡約定理を容易に示すことができるようになった．

構築した型体系に対して，いくつかの例題を用いてその妥当性を確認した．プログラム理論の点からも重要な例題として，型無しラムダ計算のハイパーグラフ書換えへのエンコーディング手法を新たに開発して，それが型付け可能であることを示した．

(4) グラフの形状を扱うユーザ定義型の開発

上述のケイパビリティに基づく型体系と異なる役割を担う型体系として，グラフの形状を扱う型体系も理論と実用の両面から重要である．関数型言語で扱う通常の型は本質的に木構造であるが，グラフの型は共有や循環等を含む複雑な構造を扱う必要がある．グラフの型に関する研究は過去にも存在するが，実際的なプログラミング言語と結びついた研究はなかった．

LMNtal においてはグラフの型は二つの役割を果たす．一つはグラフ書換えルールの記述力の向上のための動的な型で，もう一つはプログラム理解と解析のための静的な型である．以下この二つについて順に述べる．

一つめの動的な型とは，LMNtal 処理系が提供している組込みの型の概念を一般化するものである．LMNtal の書換えルールは，左辺でマッチすべき部分グラフを具体的に指定するか，連結グラフや整数アトムなどあらかじめ定められた一定の性質を満たす部分グラフを包括的に指定することを許していたが，部分グラフの族を利用者が自由に定義することはこれまでできなかった．そこで，グラフの集合を表現するための再帰的な生成文法を設計した．この生成文法は，論理型言語における重要データ構造である差分リストや，抽象構文で表した数式の評価順序を記述するのに用いる評価文脈など，複数のルート（出入口）をもつ構造（パターン）を表現する能力をもつ．さらに，利用者が定義したパターンを用いてグラフ書換えの簡約基を指定できる試作処理系を実装して公開した．パターン記述言語の表現力の確認のために，評価文脈を用いたプログラミング言語の操作的意味論や一級継続の意味論をグラフ書換え系として記述し，実行することに成功した．また，実装にあたって，パターンマッチングのメモ化が多くの例題で計算量を改善することを確認した．一方，汎用性の高いメモ化アルゴリズムの設計と実装は今後の課題となった．

次に，グラフ構造を扱う静的型体系として提案された Shape Type を LMNtal に導入した．導入した型体系は，二分木のすべての葉を線形リストで連結した構造など，他のプログラミング言語では定義困難なデータ構造を扱うことができる．グラフが特定の構造を持っているかどうかの検査（グラフの型検査）および書換えルールがその構造を崩さないかどうかの検査（ルールの型検査）の概念を定め，前者に関してはアルゴリズムを提示し，その停止性と健全性の証明を与えた．またグラフやルールの型検査がグラフ書換えを用いた探索に帰着でき，LMNtal プログラムの非決定的な実行を探索に利用できることに着目し，既存の LMNtal 処理系を用いて検査を行う手法を実装した．

(5) 処理系の整備と公開

本研究期間および将来のコンパイラ発展のための最初のステップとして，運用中の LMNtal コンパイラのコードレビューを行い，今後の発展のために必要な本質的部分の抽出を行った．コンパイラはこれまで，LMNtal がモデル検査用言語として発展する以前の実行時処理系等と一体のソフトウェアとして提供されていたが，これを大幅に整理して，新たなコンパイラ発展のベースとなるコンパイラを構築した．

次に，4.(1)で述べたリンクの互換に基づくグラフ書換えを実現する抽象機械命令列

を生成するコンパイラの整備を進めた．また新たに導入した抽象機械命令を実行時処理系 slim に実装した．

また，HyperLMNtal の整備の一環として，ハイパーリンクに対して属性が付与できるように言語機能の拡張を行った．この機能を利用するためにコンパイラと実行時処理系の双方を整備するとともに，HyperLMNtal の言語仕様と処理系の妥当性評価のために，多相ラムダ計算の型体系および高階単一化機能をもつ高階論理型言語の HyperLMNtal へのエンコーディングを行い，動作確認とそれを通じた処理系改良を行った．

LMNtal コンパイラはこれまでオープンソースソフトウェアではなかったが，上記を含む整備改良の諸成果を統合したコンパイラを，新たに GitHub からオープンソースソフトウェアとして公開した．

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔学会発表〕（計 8 件）

① 吉元 佑介，上田 和紀：グラフ書換え言語 LMNtal への Shape Type の導入と実装，第 17 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（PPL2015），道後プリンスホテル（愛媛県松山市），2015 年 03 月 04 日

② Alimujiang Yasen, Kazunori Ueda: Implementing L-lambda in HyperLMNtal, 12th Asian Symposium on Programming Languages and Systems (APLAS 2014), 2014 年 11 月 17 日, National Univeristy of Singapore (Singapore)

③ Alimujiang Yasen, Kazunori Ueda: Implementing a subset of Lambda Prolog in HyperLMNtal，日本ソフトウェア科学会 第 31 回大会，2014 年 09 月 10 日，名古屋大学東山キャンパス（愛知県名古屋市）

④ 奈良 耕太，上田 和紀：パターン定義によるマッチングを導入したグラフ書換え言語とその実装，日本ソフトウェア科学会 第 31 回大会，2014 年 09 月 08 日，名古屋大学東山キャンパス（愛知県名古屋市）

⑤ Alimujiang Yasen，上田和紀：Encoding type systems into HyperLMNtal，日本ソフトウェア科学会第 30 回大会 (JSSST2013)，2013 年 09 月 11 日，東京大学（東京都文京区）

⑥ Yuuki Shinobu, Yoshinori Tanabe, Kazunori Ueda: Formalization of the Graph Rewriting Operations of LMNtal by Coq, 11th Asian Symposium on Programming Languages and Systems (APLAS 2013)，2013 年 12 月 10 日，Melbourne (Australia)

⑦ 信夫裕貴，田辺良則，上田和紀：LMNtal におけるグラフ書換え操作の Coq による形式化，日本ソフトウェア科学会第 30 回大会（JSSST2013），2013 年 09 月 13 日，東京大学（東京都文京区）

⑧ 信夫 裕貴，上田 和紀： LMNtal コンパイラの検証に向けたグラフ書き換え操作の形式化，第 15 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（PPL2013），2013 年 03 月 04 日，御宿東鳳（福島県会津若松市）

〔図書〕（計１件）

① Kazunori Ueda, Towards a Substrate Framework of Computation. In *Concurrent Objects and Beyond*, Gul Agha et al. (eds.), LNCS 8665, Springer-Verlag, 2014, pp.341-366，査読有，
DOI:10.1007/978-3-662-44471-9_15

〔その他〕
ホームページ等

① LMNtal: モデル検査機能と統合ビジュアル環境を備えた階層グラフ書換え言語処理系
http://www.ueda.info.waseda.ac.jp/lmntal/

② LMNtalコンパイラ公開ページ
https://github.com/lmntal/lmntal-compiler/

③ Context-sensitive FlatLMNtal
http://www.ueda.info.waseda.ac.jp/~nara/csflatlmn

６．研究組織
(1)研究代表者
上田 和紀（UEDA, Kazunori）
早稲田大学・理工学術院・教授
研究者番号：１０２５７２０６